

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

«ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

Факультет

Вычислительной техники

Кафедра

Вычислительная техника

Направление подготовки 09.03.01 «Информатика и вычислительная техника»

Профиль «Программное обеспечение средств ВТ и
автоматизированных систем»

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ
РАБОТА БАКАЛАВРА

на тему

Моделирование алгоритма WRED для управления очередями в
коммутаторах

Студент

 03.06.20
(подпись, дата)

Жидкова Карина Олеговна
(ФИО полностью)

Руководитель


(подпись, дата)

Н.Н. Коннов
(фамилия, инициалы)

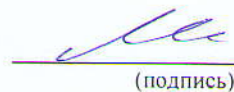
Нормоконтролёр


(подпись, дата)

А.В. Кучин
(фамилия, инициалы)

Работа допущена к защите (протокол зас. кафедры от 03.06.2020 г. № 12)

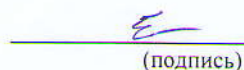
Заведующий кафедрой


(подпись)

Митрохин М.А.

Работа защищена с отметкой хорошо (протокол заседания ГЭК от
23.06.2020 № 14)

Секретарь ГЭК


(подпись)

Гурин Е.И.

Пенза, 2020

Федеральное государственное бюджетное образовательное учреждение
высшего образования
ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ПОЛИТЕХНИЧЕСКИЙ ИНСТИТУТ

Факультет
Вычислительной техники

Кафедра
Вычислительная техника

«Утверждаю»
Заведующий кафедрой
 М.А. Митрохин
«30» апреля 2020 г.

ЗАДАНИЕ
НА ВЫПУСКНУЮ КВАЛИФИКАЦИОННУЮ РАБОТУ
БАКАЛАВРА

1. Студент Жидкова Карина Олеговна гр. 16BB2 факультета вычислительной техники.

2. Тема работы: «Моделирование алгоритма WRED для управления очередями в коммутаторах».

Тема утверждена приказом ПГУ № 0335а/о от 30.04.2020 г.

3. Руководитель работы к.т.н, профессор кафедры «ВТ» Коннов Н.Н.

4. Задание на работу (назначение разработки, исходные данные и т.п.).

Разработать модели взвешенного алгоритма произвольного раннего обслуживания очередями (WRED) в коммутаторе Ethernet, обеспечивающего возможность исследование вероятностных характеристик продвигаемого трафика в зависимости от интенсивности нагрузки и структуры поступающего трафика

5. Перечень подлежащих разработке вопросов.

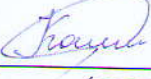
Требуется произвести анализ предметных областей: исследовать алгоритмы управления очередями, такие как RED и WRED с полной буферизацией и при буферизации «на лету».

6. Календарный график выполнения работы

№ п/п	Наименование этапов работы	Объем работы	Срок выполнения	Подпись руководителя
1.	Изучение предметной области	10	30.04.2020 - 03.05.2020	
2.	Анализ требований	10	03.05.2020 - 05.05.2020	
3.	Создание имитационной модели RED – WRED при буферизации «налету»	20	05.05.2020 - 31.05.2020	
4.	Создание имитационной модели RED – WRED при полной буферизации	25	31.05.2020 - 03.06.2020	
5.	Исследование полученных моделей	25	30.04.2020 - 03.05.2020	
6.	Оформление пояснительной записки	10	03.05.2020 - 05.05.2020	

Дата выдачи задания "30" апреля 2020 г.

Руководитель бакалаврской работы  Коннов Н.Н.
(подпись, дата)

Задание к исполнению принял студент  30.04.20 Жидкова К.О.
(подпись, дата)

Работу к защите допустить
Декан факультета  Фионова Л.Р.
(подпись, дата)

РЕФЕРАТ

Выпускная квалификационная работа содержит 81 лист, 29 рисунков, 8 таблиц, 19 источников, 1 приложение.

ТЕЛЕКОММУНИКАЦИОННАЯ СЕТЬ, КОММУТАТОР, КАЧЕСТВО ОБСЛУЖИВАНИЯ QOS, ETHERNET, МЕХАНИЗМЫ QOS, УПРАВЛЕНИЕ ОЧЕРЕДЯМИ, RED, WRED.

Цель работы – разработка модели взвешенного алгоритма произвольного раннего обслуживания очередями (WRED) в коммутаторе Ethernet, обеспечивающая возможность исследование вероятностных характеристик продвигаемого трафика в зависимости от интенсивности нагрузки и структуры поступающего трафика. Обеспечение возможности моделирования алгоритмов вероятностного заблаговременного определения перезагрузки (RED) и простого пассивного алгоритма управления очередями коммутаторов (TailDrop).

В результате проделанной работы были разработаны модели алгоритмов управления очередями RED-WRED с полной буферизацией и буферизацией «налету».

					ПГУ 2.090301.14.001 ПЗ				
Изм.	Лист	№ докум.	Подпись	Дата					
Разраб.		Жидкова К.О.			<div>Моделирование алгоритма WRED для управления очередями в коммутаторах</div> <div>Пояснительная записка</div>				
Провер.		Коннов Н.Н.							
Н. Контр		Кучин А.В.							
Утверд.		Митрохин М.А.							
					Лит.	Лист	Листов		
						2	81		
					ФВТ, каф. ВТ				
					гр. 16BB2				

ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ

WRED (Weighted Random Early Detection) – взвешенный алгоритм произвольного раннего обслуживания;

RED (Random Early Detection) – алгоритм вероятностного заблаговременного определения перезагрузки;

QoS (Quality of Service) – технология предоставления различным классам трафика различных приоритетов в обслуживании – качества обслуживания.

					ПГУ 2.090301.14.001 ПЗ	Лист
						3
Изм.	Лист	№ докум.	Подпись	Дата		

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	6
1 МЕХАНИЗМЫ ОБСЛУЖИВАНИЯ В КОМПЬЮТЕРНЫХ СЕТЯХ	8
1.1 Общие сведения о качестве обслуживания	8
1.2 Механизмы реализации Qos.....	10
1.2.1 Маркирование.....	12
1.2.2 Механизмы обслуживания очередей	13
1.2.3 Управление очередью.....	14
1.2.4 Управление полосой пропускания	14
2 ОСОБЕННОСТИ АЛГОРИТМОВ УПРАВЛЕНИЯ ОЧЕРЕДЯМИ.....	17
2.1 Классификация алгоритмов управления очередями	17
2.2 Математическое описание алгоритма RED.....	19
2.3 Математическое описание алгоритма WRED	22
3 МОДЕЛИРОВАНИЕ АЛГОРИТМОВ УПРАВЛЕНИЯ ОЧЕРЕДЯМИ	24
3.1 Выбор аппарата для имитационного моделирования алгоритмов управления очередями	24
3.2 Обобщенная структура модели управления очередями	27
3.3 Описание универсальной модели алгоритмов управления очередью RED – WRED при полной буферизации.	39
3.4 Описание универсальной модели алгоритмов управления очередью RED – WRED при буферизации «налету»	46
4 ЭКСПЕРИМЕНТАЛЬНОЕ ИССЛЕДОВАНИЕ АЛГОРИТМОВ УПРАВЛЕНИЯ ОЧЕРЕДЯМИ RED – WRED	49
4.1 Исходные данные для моделирования.	49
4.2 Обработка результатов экспериментов	51
4.3 Анализ результатов экспериментов.....	57
5 БЕЗОПАСНОСТЬ ЖИЗНЕДЕЯТЕЛЬНОСТИ	64
5.1 Введение	64
5.2 Требования к уровню шума и вибрации	64

5.3	Требования к освещению помещений и рабочих мест.....	65
5.4	Требования к организации и оборудованию рабочих мест	65
ЗАКЛЮЧЕНИЕ		68
СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ		69
Приложение А Презентация		72

ВВЕДЕНИЕ

Современный мир тяжело представить без такой сети как Интернет. Неотъемлемой частью жизни и деятельности большинства людей стали мессенджеры, видеоконференции, электронная почта, IP-телефония, радиотрансляции, видеоконференции, интерактивные обучающие программы.

Так каким же образом такая сеть, как Интернет, в большей части которой лежит инфраструктура сетей с пакетной коммутацией, справляется с бурным ростом объемов передаваемой информации, обеспечивая при этом должный уровень передачи данных для приложений с различными требованиями к качеству обслуживания (Quality of Service – QoS). Для решения задач подобного рода были разработаны методы обеспечения качества обслуживания.

Качество обслуживания (QoS) сегодня является одним из постоянно растущих требований к сети. Введение функции качества обслуживания позволяет в современных сетях обеспечить гарантированный и дифференцированный уровень обслуживания сетевого трафика, запрашиваемого теми или иными приложениями на основе различных механизмов распределения сетевых ресурсов (ограничения интенсивности трафика, обработки очередей и приоритизации).

Заданный уровень QoS может потребоваться для ряда сетевых приложений, в частности:

- потоковые мультимедиа-приложения требуют гарантированную пропускную способность канала;
- интернет – телефония и видеоконференция требуют небольших значений задержки и ее вариации;

— телеуправление, а также некоторые направления телемедицины требуют гарантированного уровня надёжности.

					ПГУ 2.090301.14.001 ПЗ	Лист
						7
Изм.	Лист	№ докум.	Подпись	Дата		

1 МЕХАНИЗМЫ ОБСЛУЖИВАНИЯ В КОМПЬЮТЕРНЫХ СЕТЯХ

Для поддержки передачи по одной сети трафика потоковых мультимедийных приложений, видеоконференции, онлайн игры и др.) и трафика данных с различными требованиями к пропускной способности необходимы механизмы, реализуемые телекоммуникационным оборудованием для дифференцированной обработки сетевого трафика различного типа с целью обеспечения вероятностно-временных характеристик для отдельных типов трафика в зависимости от предъявляемых к ним требований.

1.1 Общие сведения о качестве обслуживания

В настоящее время большое внимание уделяется качеству обслуживания сети (Quality of Service, QoS). Термин «качество обслуживания» означает способность коммуникационной системы обеспечивать то или иное качество услуг в зависимости от вида передаваемых данных.

Введение функции качества обслуживания позволяет в современных сетях обеспечить гарантированный и дифференцированный уровень обслуживания сетевого трафика, запрашиваемого теми или иными приложениями на основе различных механизмов распределения сетевых ресурсов (ограничения интенсивности трафика, обработки очередей и приоритизации).

Например, при передаче голоса или видео по коммуникационной сети необходимо обеспечить приоритетное продвижение таких пакетов, причем без потерь информации, с тем чтобы на приемном конце не ощущалось «выпадения» кадров изображения или прерывания голоса собеседника. Таким образом, целью качества обслуживания состоит в разработке методов качественной передачи трафика через сеть, для того чтобы обеспечить комфортную доставку трафика до конечного пользователя.

Изм.	Лист	№ докум.	Подпись	Дата

Одной из самых важных характеристик качества обслуживания является задержка доставки пакета. Задержка доставки пакета является статистической (вероятностной) величиной. Она состоит из двух составляющих: предсказуемой и непредсказуемой. Предсказуемая составляющая задержки определяется обработкой пакета, и зависит от его размера. Данная составляющая задержки не может быть минимизирована. Интерес же представляет непредсказуемая составляющая задержки пакета – джиттер, т.к. она определяется используемыми в оборудовании механизмами обеспечения качества обслуживания. В связи с этим качество обслуживания совершенствуется за счет реализации новых эффективных алгоритмов обслуживания и управления очередями, являющихся основой для предоставления гарантированного качества услуг связи. Это, естественно, требует проведения предварительного моделирования телекоммуникационных сетей с целью создания наиболее адекватной системы, способной соответствовать заданным и подразумеваемым требованиям. В связи с этим актуальной является разработка моделей с обслуживанием очередей в коммутаторе Ethernet.

В литературе, описывающей реализацию поддержки качества обслуживания в телекоммуникационных сетях, описываются следующие механизмы, используемые для обеспечения заданного качества сервиса в сетях передачи данных:

- классификация трафика;
- маркирование трафика;
- обслуживание очередей;
- управление очередью;
- управления полосой пропускания.

Функционально классификация и маркирование трафика чаще всего обеспечиваются на входных портах оборудования, а управление и обслуживание очередей, управление полосой пропускания – на выходных.

В сетевом оборудовании для обеспечения заданного уровня QoS одновременно используются все механизмы. Исключением являются механизмы маркирования и управления полосой, так как данные механизмы в основном работают на границах сети и могут быть представлены в маршрутизаторах, входящих в вычислительное «облако».

В зависимости от типа трафика необходимо комбинировать различные механизмы для обеспечения наилучшей производительности телекоммуникационного оборудования. В связи с этим любой показатель, характеризующий производительность, зависит от используемых механизмов.

1.2 Механизмы реализации Qos

Существуют следующие механизмы, используемые для обеспечения заданного качества сервиса в сетях передачи данных:

- механизмы маркирования;
- механизмы обслуживания очередей;
- механизмы управления очередями;
- механизмы управления полосой пропускания.

В сетевом оборудовании для обеспечения заданного уровня QoS одновременно используются несколько механизмов из списка, указанного выше. В зависимости от типа трафика необходимо комбинировать различные механизмы для обеспечения наилучшей производительности телекоммуникационного оборудования. В связи с этим любой показатель характеризующий производительность, зависит от используемых механизмов.

Для обеспечения заданного уровня QoS в сетевом оборудовании одновременно используются несколько механизмов, из списка, указанного выше. В зависимости от трафика необходимо совмещать различные механизмы для обеспечения наиболее высокой производительности

телекоммуникационного оборудования. В связи с этим любой показатель характеризующий производительность, зависит от используемых механизмов.

На рисунке 1 представлена структура коммутатора с поддержкой качества обслуживания.

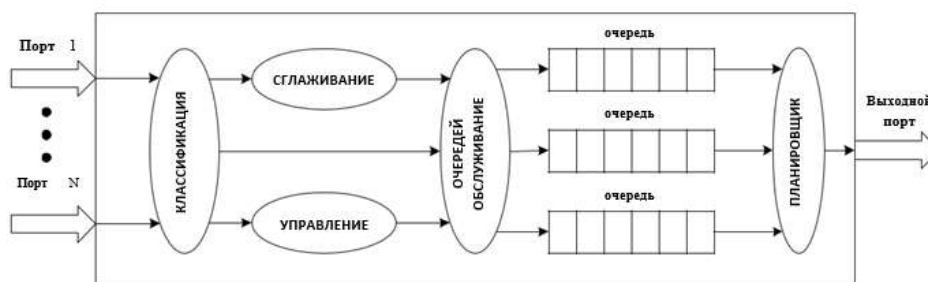


Рисунок 1 — Структура коммутатора с поддержкой качества обслуживания

Кадры входящего трафика поступают на входные порты коммутатора, далее проходят классификацию обычно, либо на основе типа класса, которому соответствует кадр. Затем, в соответствии с присвоенным классом, кадры Ethernet могут проходить через функцию сглаживания или через функцию управления нагрузкой, которая определяет, какое количество кадров сетевого трафика может быть пропущено для дальнейшей передачи для каждого класса. После пакеты перемещаются в очереди, которые могут быть определены на каждый поток или на каждый класс. Каждой очереди соответствует свой механизм управления очередями, например, RED, при помощи которого определяется, поместить ли текущий кадр в очередь или же отбросить.

Механизмы управления очередями могут использоваться также для заблаговременного сброса или маркировки кадров, чтобы заранее предупредить момент времени повышения нагрузки и не допустить переполнения очереди.

1.2.1 Маркирование

Граничные маршрутизаторы сети, поддерживающей качество сервиса, по определенным правилам классифицируют входной трафик, выполняют нормализацию и профилирование трафика, что предусматривает измерение его параметров, проверку соответствия заданным в SLA, при этом пакеты, не укладывающиеся в рамки установленных правил, могут быть отсеяны), маркируют трафик и передают транзитным устройствам для дальнейшего продвижения по сети. В результате однотипным приложениям соответствует одинаковый приоритет, и соответствующие им пакеты обрабатываются одинаково. Для маркировки пакетов (кадров) используется поле типа сервиса. Базируясь на значении поля типа сервиса, транзитный маршрутизатор (коммутатор) помещает пакет в очередь, соответствующую заданному приоритету. В результате однотипным приложениям соответствует одинаковый приоритет, и обрабатываются соответствующие пакеты сходным образом.

Для маркировки пакетов используется поле типа сервиса. Базируясь на значении поля типа сервиса, транзитный маршрутизатор помещает пакет в очередь, соответствующую заданному приоритету.

Достоинства механизма маркирования состоят в том, что маркирование, во-первых, обеспечивает единое понимание того, как должен обрабатываться трафик определенного класса, а во-вторых, позволяет разделить весь трафик на относительно небольшое число классов и не анализировать каждый информационный поток отдельно.

Хотя маркирование не гарантирует качество обслуживания на 100%, у него есть серьезные преимущества. Например, нет необходимости в организации предварительного соединения и в резервировании ресурсов. А так как в механизме маркирования используется небольшое, фиксированное количество классов и трафик абонентов распределяется по общим очередям, то не требуется высокая производительность сетевого оборудования.

Изм.	Лист	№ докум.	Подпись	Дата

Данный механизм предлагает базовый уровень, обеспечивающий поддержку качества обслуживания, является гибким механизмом для разработчиков активного сетевого оборудования.

Однако при использовании данной архитектуры возникают определенные сложности.

Каждый поставщик сетевых услуг обеспечивает качество сервиса в своей «зоне ответственности», однако нет гарантии, что поддержка окажется сквозной.

1.2.2 Механизмы обслуживания очередей

Механизмы обслуживания очередей отвечают за формирование выходного трафика, при наличии нескольких очередей.

Существует несколько основных механизмов обслуживания очередей, эффективное применение которых обусловлено характером входного трафика. Часто используются следующие механизмы обслуживания очередей:

- приоритетный (FPP);
- циклический (RR);
- взвешенные справедливый (WFQ).

Приоритетный механизм обслуживания работает следующим образом. Каждой очереди присваивается значение приоритета. Сначала обслуживается очередь с наибольшим приоритетом и так далее. При этом приоритетная очередь «захватывает» выходной порт телекоммуникационного оборудования. Данный механизм предпочтителен при передаче трафика реального времени.

Циклическое обслуживание очередей гарантирует, что каждая очередь будет обслужена в определенном объеме. Объем зависит от полосы, выделенной для передачи трафика определенного класса.

Взвешенное справедливое обслуживание обеспечивает равномерное

распределение полосы пропускания между всеми очередями. Отличием данного механизма от циклического обслуживания очередей является то, что взвешенное справедливое обслуживание не зависит от распределения полосы пропускания, в связи с чем без применения механизмов управления полосой пропускания все очереди передают равное количество трафика.

1.2.3 Управление очередью

Механизмы управления очередью отвечают за выполнение следующих действий:

- выбор очередного пакета из очереди для обработки;
- выбор пакета, который будет удален из очереди при её переполнении.

Использование управления очередями тесно взаимосвязано с обслуживанием очередей, вследствие чего механизмы управления очередями часто рассматривают как подмножество механизмов обслуживания очередей. Включение механизмов управления очередями в механизмы обслуживания очередей не является корректным, так как обслуживание очередей определяет порядок обслуживания нескольких очередей, а управление очередью связано с работой одной очереди. Комбинация различных алгоритмов этих двух механизмов дают различный результат при одном и том же входном трафике.

1.2.4 Управление полосой пропускания

Данные механизмы реализуют резервирование определенной доли сетевых ресурсов, необходимой информационному потоку обмена данными. Применение управления полосой пропускания наиболее критично, при передаче трафика реального времени.

Появление данных механизмов связано с тем, что традиционно трафик передавался по методу «best effort» – «с максимальными усилиями». То есть

сеть старается обработать поступающий трафик как можно быстрее, но при этом никаких гарантий относительно результата своих усилий не дает. Не гарантируется ни проверка сети обеспечить поток данных сетевыми ресурсами, ни приоритизация.

Другими словами, безотносительно к какому типу трафика относятся информационные пакеты, они обрабатываются по принципу «первый пришел - первый получил обслуживание». Очевидно, что эта модель не подходит для передачи трафика со специфическими требованиями к задержке, производительности или надежности передачи данных. Для перехода к новым приложениям на сетях с коммутацией пакетов нужен свой механизм обеспечения качества передачи. В такой структуре сети пакеты, чувствительные к задержке, должны отличаться от пакетов данных, не требующих срочной обработки, и это отличие должно фиксироваться узлами сети.

Хотя причины обеспечения качества услуг в сетях взаимосвязаны, можно выделить четыре основные составляющие, значения которых можно определить независимо. Эти параметры в настоящее время могут контролироваться оборудованием сетей, и они чаще всего включаются в соглашения провайдера услуг с клиентом о заданном уровне качества SLA (Service Level Agreement). В этом договоре сервис-провайдер и потребитель определяют:

- параметры качества обслуживания трафика, которые интересуют потребителя и которые согласен поддерживать провайдер — например, средняя пропускная способность, максимальные задержки и вариации задержек, максимальная интенсивность потерь данных, коэффициент готовности сервиса, максимальное время восстановления сервиса после отказа;
- методы измерения этих параметров;
- определение платы за обслуживание. Система оплаты может быть достаточно сложной, особенно если соглашение предусматривает

-
- несколько уровней качества обслуживания, которые оплачиваются по разным тарифам;
- санкции за нарушение обязательств провайдера по обеспечению надлежащего качества обслуживания, а также за отклонение параметров трафика пользователя от оговоренных значений. Эти санкции могут выражаться в виде штрафов, либо в иной форме, например, в форме предоставления сервиса в течении некоторого времени бесплатно или по пониженному тарифу;
- как и любой договор, соглашение SLA по взаимному согласию провайдера и клиента может включать большое количество различных дополнительных статей. Например, статью, оговаривающую условия перехода к более качественному обслуживанию или обслуживанию с
- разным уровнем качества в зависимости от дня недели или времени суток;
- соглашение может включать также правила «кондиционирования» пользовательского трафика, то есть обработки трафика, который выходит за оговоренные пределы. Также правила «кондиционирования» могут определять условия отбрасывания или маркировки пакетов-нарушителей (помеченные пакеты будут отбрасываться сетью не всегда, а только в том случае, когда сетевые устройства испытывают перегрузки).

2 ОСОБЕННОСТИ АЛГОРИТМОВ УПРАВЛЕНИЯ ОЧЕРЕДЯМИ

2.1 Классификация алгоритмов управления очередями

После классификации пакетов и помещения их в очередь, соответствующую классу, они ждут своей обработки устройством обслуживания очередей (планировщиком). Существует несколько основных алгоритмов обслуживания очередей:

- приоритетного обслуживания;
- взвешенного обслуживания;
- комбинированного обслуживания.

В алгоритме FIFO (First In – First Out) все поступившие пакеты в случае перегрузки ставятся в одну очередь и обслуживаются в порядке поступления. В чрезмерной простоте и есть главный недостаток, так как пакеты чувствительного к задержкам трафика могут быть надолго задержаны пульсациями эластичного трафика.

Алгоритм приоритетного обслуживания применяется для преимущественной обработки одного класса трафика по сравнению с другими. Механизм приоритетного обслуживания основан на разделении всего сетевого трафика на небольшое количество классов и назначение каждому из них приоритета.

В сетевом устройстве, поддерживающем приоритетное обслуживание, имеется несколько очередей (буферов), по одной для каждого класса.

В период перегрузок пакет помещается в очередь, которая соответствует его классу. Планировщик (обслуживающее устройство) выбирает из очередей пакеты по следующей схеме: если в более приоритетной очереди появляются пакеты, то планировщик в первую очередь передает их в выходную очередь до тех пор, пока в более приоритетной очереди не закончатся все пакеты.

Далее планировщик переходит к обслуживанию менее приоритетной очереди. Поэтому пакеты с низким приоритетом обрабатываются только в случае, когда пусты все более приоритетные очереди.

Приоритетное обслуживание очередей обеспечивает высокое качество обслуживания для пакетов из самой приоритетной очереди. Пакеты высоко приоритетного потока всегда получают, ту пропускную способность, которая им необходима. Уровень задержек таких пакетов минимален, но не нулевой, так как зависит от интенсивности потока и размера пульсации. Чем выше эти параметры, тем вероятнее возникновение очереди, образованной пакетами этого потока. Трафик всех остальных приоритетных классов почти прозрачен для трафика самого приоритетного класса. Качество их обслуживания будет ниже, при этом уровень снижения может быть довольно существенным и зависит от характеристик высоко приоритетного трафика. Например, в период высокой интенсивности трафика высшего приоритета, обслуживание остального может приостановиться, поэтому приоритетное обслуживание обычно применяется для класса трафика, чувствительного к задержкам, но имеющего небольшую интенсивность (например, телефонии).

Алгоритм взвешенного обслуживания разработан с целью предоставления всем классам трафика определенного минимума пропускной способности. Под весом класса понимается процент предоставляемой ему пропускной способности от полной пропускной способности ресурса, предоставляемый в случае перегрузок (для входного потока таким ресурсом является процессор, а для потока после выполнения коммутации -выходной интерфейс). Трафик делится на несколько классов и для каждого класса выделяется отдельная очередь для пакетов.

Очереди обслуживаются последовательно и циклически, и в каждом цикле обслуживания из каждой очереди выбирается такой объем байтов, который соответствует весу данной очереди. Допустим, у нас имеются четыре очереди, которым выделено 10%, 20%, 30% и 40% от пропускной способности интерфейса соответственно.

Если цикл просмотра очередей равен одной секунде, а скорость выходного интерфейса составляет 100 Мбит/с, то в случае перегрузки первой очереди выделяется 100 мс времени и выбирается 10 Мбит данных, из второй — 20 Мбит, из третьей — 30 Мбит и из четвертой — 40 Мбит.

В результате каждому классу предоставляется гарантированный минимум пропускной способности, что является более желательным результатом по сравнению с подавлением трафика низших классов в алгоритме приоритетного обслуживания. Так как данные выбираются пакетами, а не байтами, то реальное распределение пропускной способности между классами будет несколько отличаться от заданного. Чем больше время цикла, тем точнее соблюдаются соотношения между классами трафика, так как влияние отдельных пакетов уменьшается. С другой стороны, длительный цикл приводит к большим задержкам при передаче, потому что планировщик возвращается к каждой очереди не чаще чем один раз за цикл (в нашем примере — не чаще одного раза в секунду). Поэтому при выборе времени цикла не следует забывать про задержки, которые могут возникнуть в этом случае.

На уровень задержек и вариаций задержек пакетов для каждого отдельного класса трафика при взвешенном обслуживании в большой степени влияет коэффициент использования. В этом случае коэффициент использования подсчитывается как отношение интенсивности входного трафика конкретного класса к пропускной способности, выделенной для этого же класса.

2.2 Математическое описание алгоритма RED

Алгоритмы активного управления очередями при разработке были ориентированы на соединения TCP в связи с тем, что именно в протоколе TCP реализованы механизмы регулирования нагрузки в зависимости от состояния сети. Реализация алгоритмов активного управления очередями позволяет добиться ряда положительных моментов:

возможность избежать проблем захвата очереди пакетами одного или нескольких потоков;

- решить проблему отсутствия возможности заблаговременного обнаружения перегрузки;
- обеспечить низкую задержку пакета в буфере;
- увеличить коэффициент использования сетевых ресурсов.

Алгоритм управления очередями «вероятностное заблаговременное определение перезагрузки» (Random Early Detection, далее – RED) позволяет контролировать нагрузку в рамках очереди маршрутизатора и при обнаружении перегрузки или состояния близкого к перегрузке осуществлять вероятностный сброс пакетов. Сброс пакета осуществляется на базе вычисления вероятности, поэтому:

- можно соблюдать принцип «справедливого распределения ресурсов», и, соответственно, избегать возникновения проблемы lock-out;
- осуществлять последовательный сброс пакетов, принадлежащих различным соединениям, т.е. избегать «глобальной синхронизации».

Алгоритм RED ориентирован на работу с протоколом TCP, поэтому сброс пакета позволит источнику нагрузки уменьшить размер окна $cwnd$ и, таким образом, понизить нагрузку.

Рассмотрим общий принцип функционирования базового алгоритма RED. Алгоритм состоит из двух частей: оценка среднего размера очереди и принятие решения о сбросе вновь поступившего пакета. При поступлении каждого нового пакета, используя фильтр высоких частот совместно с «экспоненциально взвешенной скользящей средней EWMA (low-pass filter with an exponentially moving average), RED определяет средний размер очереди avg . Далее значение avg сравнивается со значением ранее определённых границ: верхней max_th и нижней min_th , которые определяют степень нагрузки в очереди.

В случае если значение avg принадлежит интервалу $[0; min_th]$, то поступивший пакет помещается в очередь, если же значение avg принадлежит интервалу $[min_th; max_th]$, то вычисляется вероятность сброса пакета P_a осуществляется сброс поступающего пакета, либо с вероятностью $(1 - P_a)$ поступающий пакет помещается в очередь. Если же значение среднего размера очереди avg превышает значение max_th , то поступающий пакет обязательно сбрасывается. Вероятность сброса пакета является функцией от переменной avg , т.е. с повышением нагрузки повышается вероятность сброса пакета. Далее представлен метакод алгоритма.

Листинг 1. Метакод алгоритма RED:

```

Подсчёт нового значения среднего размера очереди, avg
if (avg < min_th)
  \* перегрузки нет
  accept packet
else if (min_th <= avg < max_th)
  \* близко к перегрузке, вероятностный сброс
  calculate probability Pa
  with probability Pa
  discard packet or mark packet
  else with probability (1 - Pa)
  accept packet
else if avg >= max_th
  discard packet or mark packet
  
```

На рисунке 2 представлена графическая интерпретация RED.

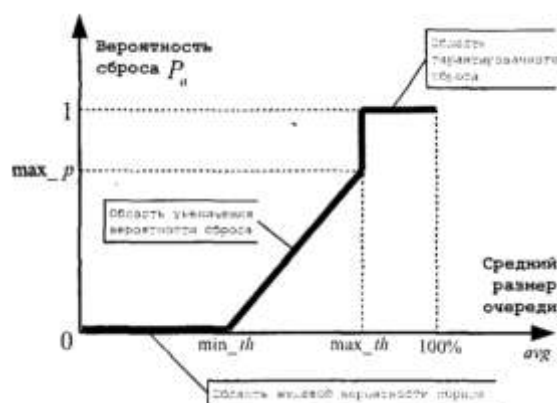


Рисунок 2 – Функция сброса алгоритма RED

Изм.	Лист	№ докум.	Подпись	Дата

RED имеет возможность осуществлять не сброс пакетов, а маркировку. Маркировка может быть произведена посредством изменения какого-либо бита в каком-либо опциональном поле в заголовке пакета с целью извещения протокола транспортного уровня, например, TCP, о надвигающейся или наступившей перезагрузке. Если в приёмнике реализован протокол, «понимающий» маркирование пакета, то протокол приёмника должен понизить размер окна *gwnd* и, тем самым, понизить количество пакетов, передаваемых в сеть за некоторый промежуток времени.

Таким образом, RED состоит из двух различных алгоритмов: с помощью алгоритма вычисления среднего размера вычисляется степень пачечности нагрузки, которая может быть принята буфером; алгоритм подсчёта вероятности маркировки или сброса пакета на основе известного значения нагрузки (т.е. значения средней длины очереди) определяет, как часто маршрутизатор маркирует или сбрасывает пакеты. Целью функционирования RED в маршрутизаторе является реализация маркировки/сброса пакетов, при которой будет соблюдаться принцип «справедливого распределения ресурсов» и избегание «глобальной синхронизации».

2.3 Математическое описание алгоритма WRED

Отличие WRED от RED состоит в том, что для пакетов каждого приоритета определён отдельный набор параметров. Например, рассмотрим случай, когда в очередь маршрутизатора поступают пакеты с двумя приоритетами – далее под пакетом с низким приоритетом будем понимать маркированный пакет, а с высоким приоритетом – обычный немаркированный пакет. Функция зависимости вероятности сброса пакета от значения среднего размера очереди ведёт себя следующим образом: при одинаковой нагрузке должен соблюдаться принцип, что вероятность сброса маркированного пакета должна превышать вероятность сброса обычного пакета.

Параметризация алгоритма WRED достаточно сложна, т.к. количество значений, которые необходимо установить, прямо пропорциональны количеству приоритетов, с которыми пакеты могут поступать в рассматриваемый маршрутизатор. Например, если учесть, что при реализации AF PHB количество приоритетов может возрасти до 12, то для полной параметризации WRED необходимо установить значения $12 * \{\min X_{tx}, \max X_{th}, \max X_p, \text{avg} X\}$ параметров, где X – номер приоритета. Частный случай функции вероятности сброса пакета для двух приоритетов алгоритма WRED представлен на рисунке 3.

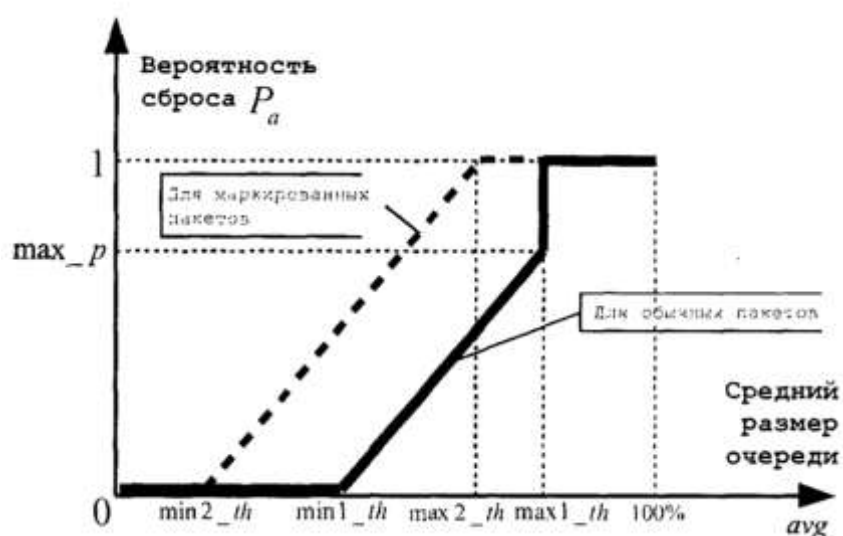


Рисунок 3 – WRED: маркировка пакетов может изменить вероятность сброса пакета

3 МОДЕЛИРОВАНИЕ АЛГОРИТМОВ УПРАВЛЕНИЯ УЧЕРЕДЯМИ

3.1 Выбор аппарата для имитационного моделирования алгоритмов управления очередями

В настоящее время существует множество программ, предназначенных для создания и исследования моделей. Основную часть систем моделирования занимают программы, использующие математический аппарат и предназначенные для сбора статистических данных. Как правило, такие системы не являются узконаправленными, а возможность моделирования является лишь одной из множества доступных функций. К таким системам можно отнести программу MatLab. Рассмотрим более подробно различные инструментальные средства моделирования.

MATLAB (сокращение от англ. «Matrix Laboratory») — пакет прикладных программ для решения задач технических вычислений и одноимённый язык программирования, используемый в этом пакете.

MATLAB представляет собой основу всего семейства продуктов MathWorks и является главным инструментом для решения широкого спектра научных и прикладных задач, в таких областях как: моделирование объектов и разработка систем управления, проектирование коммуникационных систем, обработка сигналов и изображений, измерение сигналов и тестирование, финансовое моделирование, вычислительная биология и др.

GPSS (англ. General Purpose Simulation System — общецелевая система моделирования) – общецелевая система моделирования сложных систем, использующая специфический язык программирования для имитационного моделирования различных систем, в основном систем массового обслуживания.

GPSS является языком моделирования, используемым для построения моделей и проведения моделирования на ЭВМ. Модели, описанные на языке

GPSS компактны, часто состоят из меньшего числа операторов, чем такие же модели, написанные на процедурных языках (Паскале или Си). Это объясняется тем, что в GPSS встроено максимально возможное число логических программ, необходимых для моделирующих систем. В него также входят специальные средства для описания динамического поведения систем, меняющихся во времени, причем изменение состояний происходит в дискретные моменты времени. GPSS удобен при программировании, поскольку интерпретатор GPSS многие функции выполняет автоматически (например, сбор статистических данных, описывающие поведение модели, печать итоговой статистики по завершении моделирования и т.п.). В язык включены и другие полезные элементы, например, GPSS обслуживает таймер модельного времени, планирует события, которые должны произойти позднее в течение времени моделирования, вызывает их своевременное появление и управляет очередностью поступления.

Язык моделирования дискретных систем GPSS позволяет строить модель сложной дискретной системы, через описание ее отдельных элементов и логических правил их взаимодействия в процессе работы всей системы. При этом для определенного класса моделируемых систем выделяется небольшой набор абстрактных элементов— объектов. Набор логических правил также ограничен, что позволяет ограничить его описание небольшим числом стандартных операций. Поэтому программная модель системы представляет собой комплекс программных компонентов, описывающих функционирование объектов и выполняющих логические операции над ними.

Таким образом, язык GPSS ориентирован моделирование классов объектов, процесс функционирования которых можно представить в виде множества состояний и правил перехода из одного состояния в другое, определяемых в дискретной пространственно-временной области, то есть GPSS позволяет описывать процессы с дискретными событиями.

Однако системам моделирования на GPSS свойственны недостатки, главным из которых заключается в том, что на языке GPSS достаточно сложно представить непосредственно процессы обработки данных на уровне алгоритмов. Это связано с высоким уровнем абстракций, изначально закладываемым в описание объекта моделирования. Кроме того, модель представляет собой программу, а значит не имеет графической интерпретации, что затрудняет процесс разработки модели и снижает наглядность модели в целом.

Поэтому использование системы GPSS в качестве пакета моделирования не является целесообразным в рамках данной работы.

При выборе системы моделирования для выполнения данной выпускной работы преимущество было отдано программам, использующим аппарат сетей Петри. Это обусловлено тем, что сети Петри - это универсальная алгоритмическая система, позволяющая описывать произвольные объекты.

Существует множество различных видов сетей Петри. Каждый вид сети Петри в отдельности может быть использован для разработки простых моделей. Для описания сложных моделей были разработаны специальные средства моделирования, которые соединяют в себе несколько видов сетей Петри. В качестве основного вида сети Петри в данных пакетах моделирования используются цветные сети. Это связано с тем, что цветные сети Петри являются мощным аппаратом, позволяющим создавать довольно сложные модели.

Для выполнения моделирования и исследования полученных моделей в рамках данной выпускной квалификационной работы был выбран свободно распространяемый пакет моделирования CPN Tools, использующий математический аппарат иерархических временных цветных сетей Петри. Цветные сети Петри позволяют описывать алгоритмы, зависящие от значения жетона, передаваемого в сети. Иерархические сети Петри позволяют описывать объекты любой сложности. А временные сети Петри

делают возможным моделирование динамических характеристик объектов.

Пакет CPN Tools совмещает в себе сети Петри и функциональный язык моделирования CPN ML, который является расширением языка Standard ML, и позволяет подробно исследовать различные вопросы поведения сложных компьютерных сетей.

3.2 Обобщенная структура модели управления очередями

Обобщенная структура разработанной модели буфера выходного порта представлена на рисунке 4.

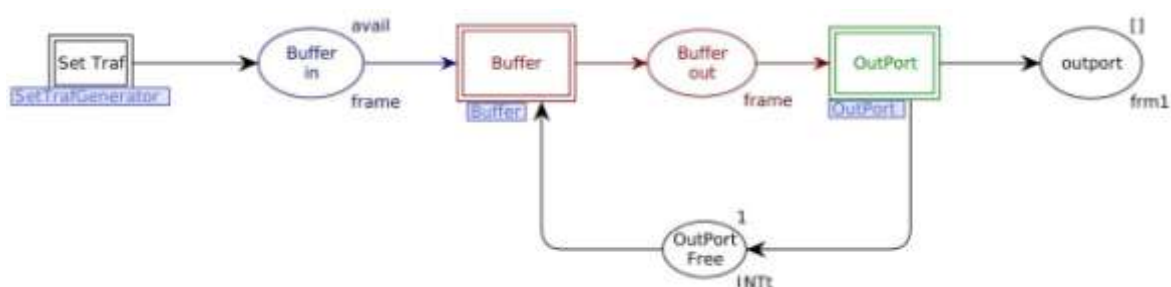


Рисунок 4 – Рабочая модель Буфера выходного торта коммутатора с ВТС на верхнем уровне иерархии

Рабочая модель включает следующие подсети:

- *SetTrafGenerator*, которая считывает из файла сгенерированные пакеты данных сети Ethernet;
- *Buffer*, который непосредственно выполняет обработку трафика, продвигаемого через порт с заданным алгоритмом диспетчеризации;
- *Buffer in* и *Buffer outport*, которые представляют собой контактные позиции, через которые передаются маркеры кадров;
- Позиция *frame* накапливает маркеры продвинутых кадров.

В отличие от традиционного для телекоммуникаций применения сетей Петри для моделирования протокола взаимодействия сетевых устройств, в нашем случае моделируется поток кадров, продвижение которых в сети определяется содержанием самих кадров, поэтому очень важно эффективно использовать аппарат цветности маркеров. Было введено два условных типа

маркеров:

1. «Информационные» маркеры, продвижение которых имитирует обработку кадра в коммутаторе:

- множество цветов *frm* отражает структуру передаваемого кадра Ethernet и анализируется на очередном этапе обработки кадра в коммутаторе;
- множество цветов *frame* представляет однонаправленный канал, который может быть либо занят передачей кадра (*frm*), либо свободен (*avail*), что позволяет выявлять и обрабатывать события, связанные с наличием или отсутствием кадра.

Маркеры с цветом *avail* позволяют обеспечить безопасность сети Петри (один маркер в любой позиции) и решить проблему отсутствия в CPN Tools ингибиторных (запрещающих) дуг.

2. «Управляющие» маркеры, цвет которых отображает состояние обработки потока кадров в коммутаторе (размер очереди, количество переданных данных, признак очистки очереди и др.).

Исходя из структуры Ethernet кадра, изложенной в разделе 1, было выбрано, чтобы цвет описывал два MAC адреса, приоритет качества обслуживания QoS, размер данных, содержащихся в кадре, время прихода кадра в сеть и время ухода кадра из сети.

В таблицах 1-3 представлены описание цветов, переменных и функций.

Таблица 1 — Описание цветов, используемых в модели (My Colors)

Описание	Комментарий
<i>colset MAC = int with 1..10 timed</i>	Используется для описания MAC-адрес источника или приемника и может принимать целочисленные значения от единицы до десяти
<i>colset QoS = int with 0..7</i>	Используется для описания приоритета кадра, который может принимать целочисленные значения.

Продолжение таблицы 1

$colset\ frm = product$ $INT * MAC * MAC * QoS * INT * INTt;$	Используется для описания структуры кадра целиком (см. раздел 1.2) и состоит из цветов, описанных выше и стандартных цветов INT , которые используются для описания соответственно номера кадра, размера кадра с учётом длины полезной нагрузки, поля адреса и управления (см. раздел 1.2). $INTt$ – описывает время прихода кадра
$colset\ frame = union\ f:frm + avail$ $timed$	Используется для описания нахождения в позиции маркера либо цвета frm , кадр, либо цвета $avail$, показывающей, что позиция пуста
$colset\ LBuffer = list\ frm$	Используется для описания очереди в буфере коммутатора
$colset\ RED = real$	Используется для описания размера текущей очереди и вероятности
$colset\ Rule = product$ $QoS * INT * INT * RED;$	Используется для описания характеристики принятия решений о сбросе пакета, включающая соответственно значение качества обслуживания (QoS), значение границ диапазонов средней скорости, в котором выполняется сброс с определённой вероятностью описание кадра пакета $\{QoS, minX_tx, maxX_th, maxp, \}$

Изм.	Лист	№ докум.	Подпись	Дата

ПГУ 2.090301.14.001 ПЗ

Лист

29

Продолжение таблицы 1

INT	Стандартный цвет, используемый для описания целочисленных переменных
INTt	Стандартный цвет, используемый для описания целочисленных переменных временных значений

Таблица 2 — Описание переменных (My Variables)

Описание	Комментарий
var r: Rule;	Переменная, описывающая передачную характеристику. Обработки пакета при помещении его в буфер: запись, сброс с определенной вероятностью, либо безусловный сброс.
var Pmax : RED;	Описывает максимальное значение сброса кадра
var W: RED;	Описывает постоянное сглаживание при подсчёте средней скорости
var max_th, min_th: INT;	Описывает диапазон, при котором выполняется сбрасывание кадров с определённой вероятностью. Минимальные диапазон, при котором выполняется сброс с определённой вероятностью
var avg, P:RED;	Среднее значение скорости, подсчитанное по алгоритму эксп-ого сглаживания

Продолжение таблицы 2

var szfrm : BitSize;	Размер кадра с учётом всех полей в битах (ссылка на страницу)
var sz, P2:INT;	Текущий размер очереди P2 – текущая вероятность, которая выдается
var number,i,n,delay,n,cp,b:INT;	number - номер кадра i – текущий размер очереди b – размер очереди, выраженный в кадрах delay – время прихода кадра в коммутатор тактах
var src, dst : MAC;	MAC адреса источника и приёмника

Таблица 3 — Описание функций (My Functions)

Описание	Комментарий
fun SetProb() = discrete (0,9)	Функция, принимающая дискретное значение от 0 до 9
fun GetTime() = IntInf.toInt(time());	Функция, фиксирующая момент передачи

Цвета непосредственно связаны со структурой Ethernet кадра.

Стандарт IEEE 802.1 Q определяет 12 возможных форматов инкапсуляции дополнительного поля в кадры Ethernet. Одним из возможных вариантов является введение тега, который передает информацию о принадлежности трафика к Ethernet II. Размер тега — 4 байта (рис 10). Он состоит из таких полей:

– Тег идентификатора протокола TPID (Tag Protocol Identifier). Размер поля — 16 бит. Указывает, какой протокол используется для тегирования. Для 802.1q используется значение 0x8100;

- Тег контроля информации TCI (Tag Control Information) - поле, инкапсулирующее в себе поля приоритета, канонического формата и идентификатора VLAN:

- Приоритет (Priority) - размер поля составляет 3 бита. Используется стандартом IEEE 802.1p для задания приоритета передаваемого трафика;

- Индикатор канонического формата CFI (Canonical Format Indicator) - размер поля составляет 1 бит. Указывает на формат MAC-адреса. 0 — канонический (кадр Ethernet), 1 — не канонический (кадр Token Ring, FDDI).

Идентификатор (VID) VLAN'a VLAN Identifier - размер поля составляет 12 бит. Указывает к какому VLAN'у принадлежит фрейм. Диапазон возможных значений VID от 0 до 4094. На рисунке 5 представлен тег стандарта IEEE 802.1 Q.



Рисунок 5 - Тег стандарта IEEE 802.1 Q

Таким образом, конечный формат кадра Ethernet представлен на рисунке 6.



Рисунок 6 - Формат кадра Ethernet с добавлением тега

Исходя из структуры Ethernet кадра, было выбрано, чтобы цвет описывал два MAC адреса, приоритет качества обслуживания QoS, размер

данных, содержащихся в кадре, время прихода кадра в сеть и время ухода кадра из сети.

Подсеть *Buffer* реализует функции работы буфера выходного порта.

"*Buffer_in*" и "*Buffer_out*" – позиции. Являются соответственно входом и выходом буфера выходного порта.

Подсеть *Buffer* представлена на рисунке 7.

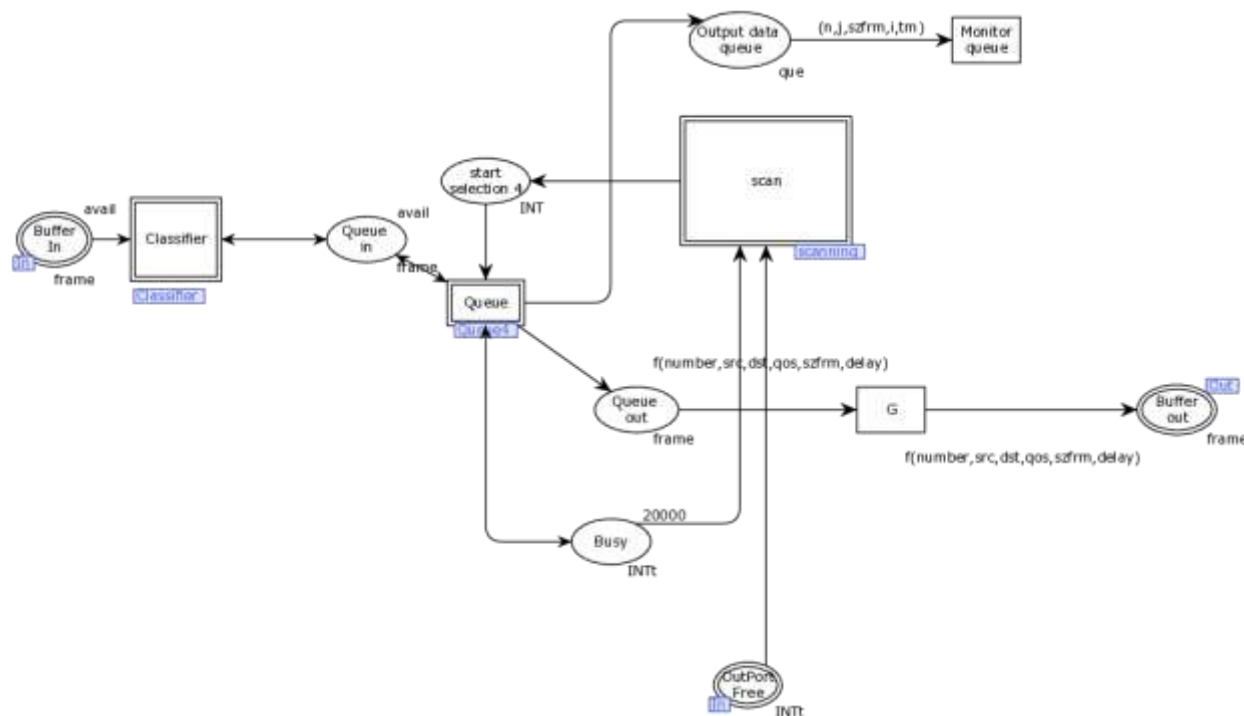


Рисунок 7 – Подсеть Buffer

Подсеть *SetTrafGenerator* описывает функцию считывания сетевого трафика из файла.

Далее представлено описание всех позиций подсети *SetTrafGenerator*:

- *Start* – стартовая позиция;
- *Read from file* - хранит фишки считанных кадров из файла в формате цвета frm1;
- *Count* – текущий номер преобразованного кадра;
- *Delta delay* – задержка между поступлениями кадров в сеть;
- *Frame output* – выходная позиция подсети.
- Далее представлено описание всех переходов подсети

SetTrafGenerator:

- *Initialise Packets* – считывание данных из файла;
- *Transfer frame* – преобразование фишки цвета *frm1* в цвет *frame* и добавление задержки.

Данные в файле хранятся в определенном виде, соответствующем структуре цвета *frm1*. Ниже представлен фрагмент такого файла.

```
1` (1,10,9,7,512, 296,0)++
1` (2,5,2,7,12096, 7389,0)++
1` (3,5,5,5,7952, 20178,0)
```

При срабатывании перехода "*Initialise Packets*" данные считываются из файла и помещаются в позицию "*Read from file*" в формате цвета *frm1*. Далее при срабатывании перехода "*Transfer frame*" в позиции "*Count*" увеличивается счетчик преобразованных фишек на единицу, в позицию "*Delta delay*" помещается фишка со значением поля *delay* текущего кадра, а берется фишка *delta* с предыдущим значением позиции.

Срабатывание перехода происходит с задержкой равной (*delay - delta*). Это сделано для того, чтобы далее в сеть кадры поступали в порядке очереди своих номеров и со своей временной задержкой. На выходе подсети *SetTrafGenerator* получаются фишки цвета *frame*, обозначающие кадры сетевого трафика.

Подсеть *SetTrafGenerator* представлена на рисунке 8.

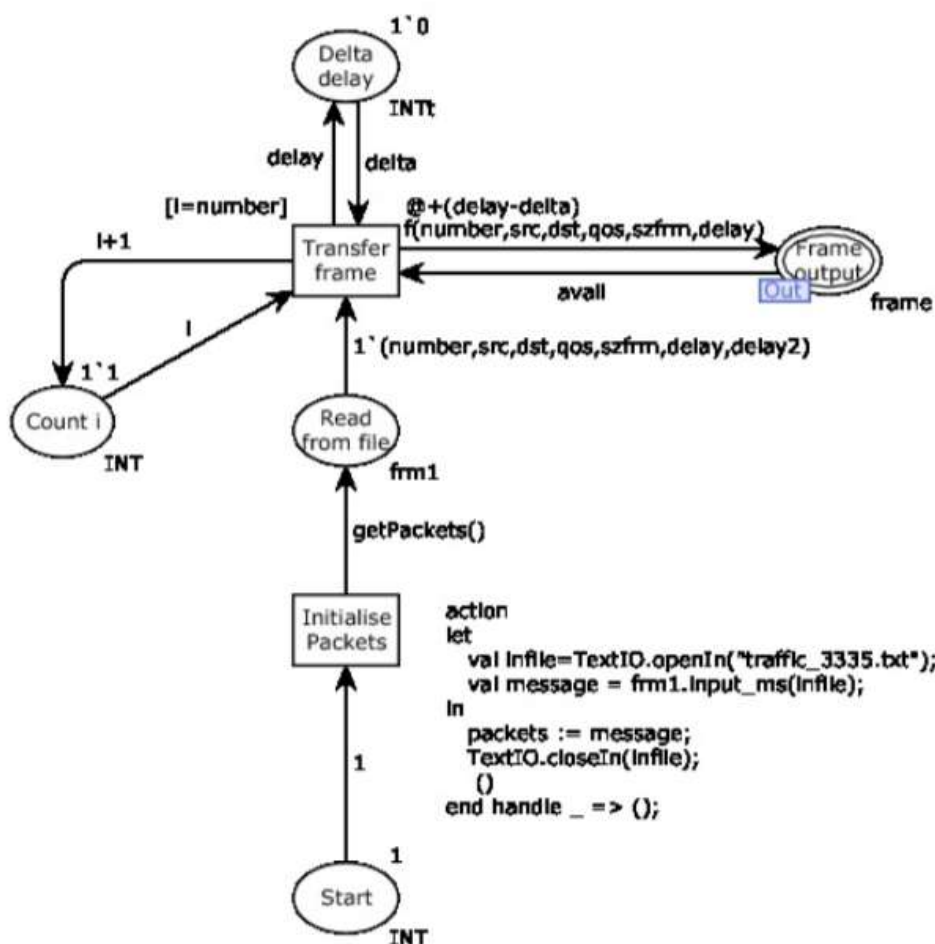


Рисунок 8 – Подсеть SetTrafGenerator

Подсеть *OutPort* реализует функции выходного порта коммутатора.

Для описания модели буфера выходного порта коммутатора использовались следующие множества цветов. Составной цвет *frm1* описывает структура кадра Ethernet и содержит в себе следующие поля: порядковый номер кадра, MAC-адреса источника и приемника, приоритет кадра, размер передаваемых данных, два поля временных задержек.

Данная подсеть включает в себя еще несколько подсетей. Ниже представлено их описание.

Далее представлено описание всех позиций и переходов подсети:

- *Buffer Out* – входная позиция подсети цвета *frame*;
- *Outport* – выходная позиция подсети цвета *frm1*;
- *OutPortFree* – позиция, которая отвечает за формирование фишки, означающей освобождение выходного порта;

– *Port* – позиция, которая отвечает за преобразование фишек из цвета *frame* в цвет *frm1*.

В данной подсети фишка цвета *frame* преобразуется в фишку цвета *frm1*, т.е. в структуру кадра добавляется еще одно поле временной задержки, которое рассчитывается, как "Текущее время - Время поступления кадра в сеть" и является временем пребывания кадра в буфере выходного порта коммутатора. При срабатывании перехода "*Port*" происходит преобразование цветов фишек и в позиции "*OutPortFree*" формируется фишка с задержкой равной $96 + \text{размер текущего кадра в битах} + 64$ ". Данная фишка является сигналом освобождения выходного порта коммутатора для следующего кадра в очереди буфера выходного порта.

Подсеть *OutPort* представлена на рисунке 9.

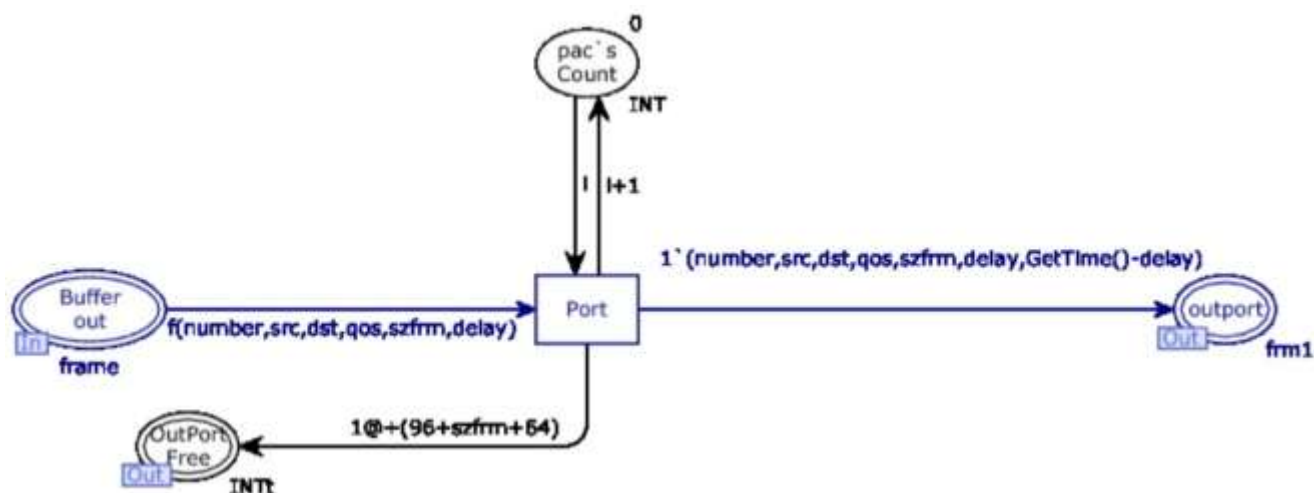


Рисунок 9 – Подсеть OutPort

Подсеть *scanning* формирует фишки выборки кадра из очереди буфера выходного порта в зависимости от наличия в очереди кадров и состояния позиции "*OutPortFree*". При наличии в очереди буфера коммутатора кадров срабатывает переход "*scan*" и с задержкой, сформированной в позиции "*OutPort Free*", формируется фишка выборки кадра из очереди.

Подсеть *scanning* представлена на рисунке 10.

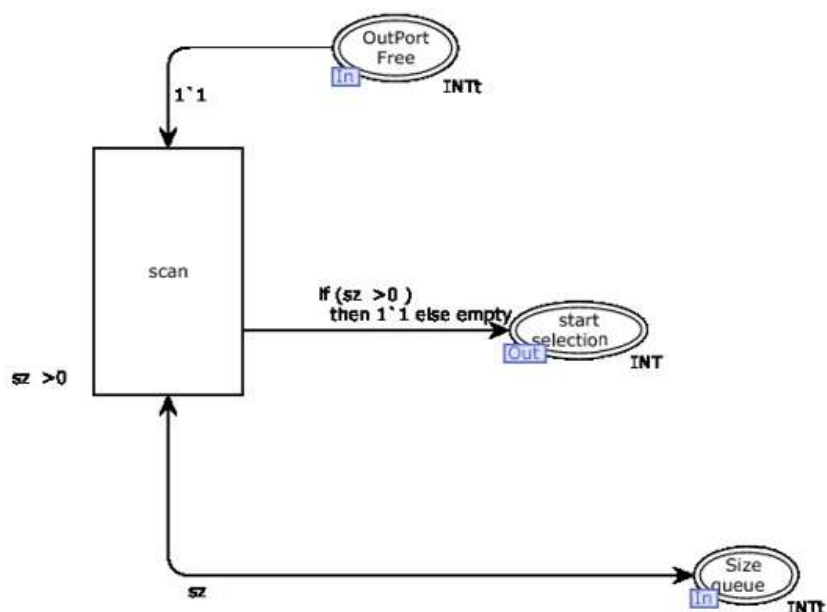


Рисунок 10 – Подсеть scanning

Подсеть Classifier классифицирует поступающие на вход буфера коммутатора кадры и в зависимости от класса кадра распределяет по каналам. В данной работе исследовались механизмы управления очередями, поэтому было принято решение оставить один канал и все кадры пускать через него.

Далее представлено описание всех позиций подсети *Classifier*:

- *Buffer In* – входная позиция подсети цвета *frame*, вход буфера выходного порта;
- *Queue in* – выходная позиция подсети цвета *frm1*, вход очереди буфера.

Далее представлено описание всех переходов подсети *Classifier*:

Classifier – классификация кадров.

В модели буфера коммутатора использовался один канал, поэтому в этой подсети фишка кадра просто передается на вход очереди буфера коммутатора. Так как в модели возможен не один канал, а несколько, была выделана данная подсеть для расширяемости системы.

Подсеть *Classifier* представлена на рисунке 11.

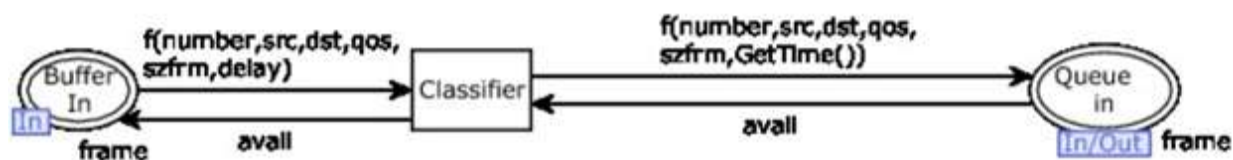


Рисунок 11 – Подсеть Classifier

Также был создан монитор, который отслеживает значение очереди в буфере коммутатора и записывает эту информацию в файл в формате "4, знак операции (1 - если пакет поступил в очередь, 0 - если пакет удалился из очереди), размер текущего кадра, размер текущей очереди, время изменения очереди".

«*Monitor queue*» учитывает и сохраняет данные о размере очереди в файл.

Монитор написан на языке моделирования CPN ML. Ниже представлен фрагмент монитора.

Листинг 2. Фрагмент монитора:

```
fun obs (bindelem) =
  let
    fun obsBindElem (Buffer'Monitor_queue (1,
      {i,j,szfrm,n,tm})) =
      Int.toString(n) ^ ", " ^ Int.toString(j) ^ ", " ^ Int.toString(szfrm)
      ^ ", " ^ Int.toString(i) ^ ", " ^ Int.toString(tm) ^ "\n"
      | obsBindElem _ = ""
  in
    obsBindElem bindelem
  end
```

Если кадр помещается в очередь буфера, то в файл записывается строка "(4, 1 , szfrm, i+szfrm, GetTime())", где переменная szfrm - это размер текущего кадра, i+szfrm - это размер очереди плюс размер текущего кадра, а GetTime() функция записи текущего времени в файл. Если же кадр выбирается из очереди буфера, то строка "(4, 0, szfrm, i-szfrm, GetTime())".

При отбрасывании кадра на входе в очередь буфера, в файл записывается строка "(0,1,0,0,0)".

Запись сгенерированного трафика в файл осуществлялась с помощью встроенных функций пакета CPN Tools мониторов. Листинг монитора на языке CPN ML представлен ниже.

Листинг 3. Монитор:

```
fun obs (bindelem) =  
  let  
    fun obsBindElem (GeneratorTraf'Write_in_fille(1, {delay1,  
      delay2, dst, i, qos,src,szfrm})) = "1`(^Int.toString(i)^",  
      ^Int.toString(src)^", ^Int.toString(dst)^",  
      ^Int.toString(qos)^", ^Int.toString(szfrm)^",  
      ^ModelTime.toString(time())^",0)++\n"  
    obsBindElem _ = ""  
  in  
    obsBindElem bindelem end.
```

3.3 Описание универсальной модели алгоритмов управления очередью RED – WRED при полной буферизации.

Алгоритм реализуется управления очередью RED – WRED при полной буферизации Подсетью Queue, которая представлена на рисунке 12.

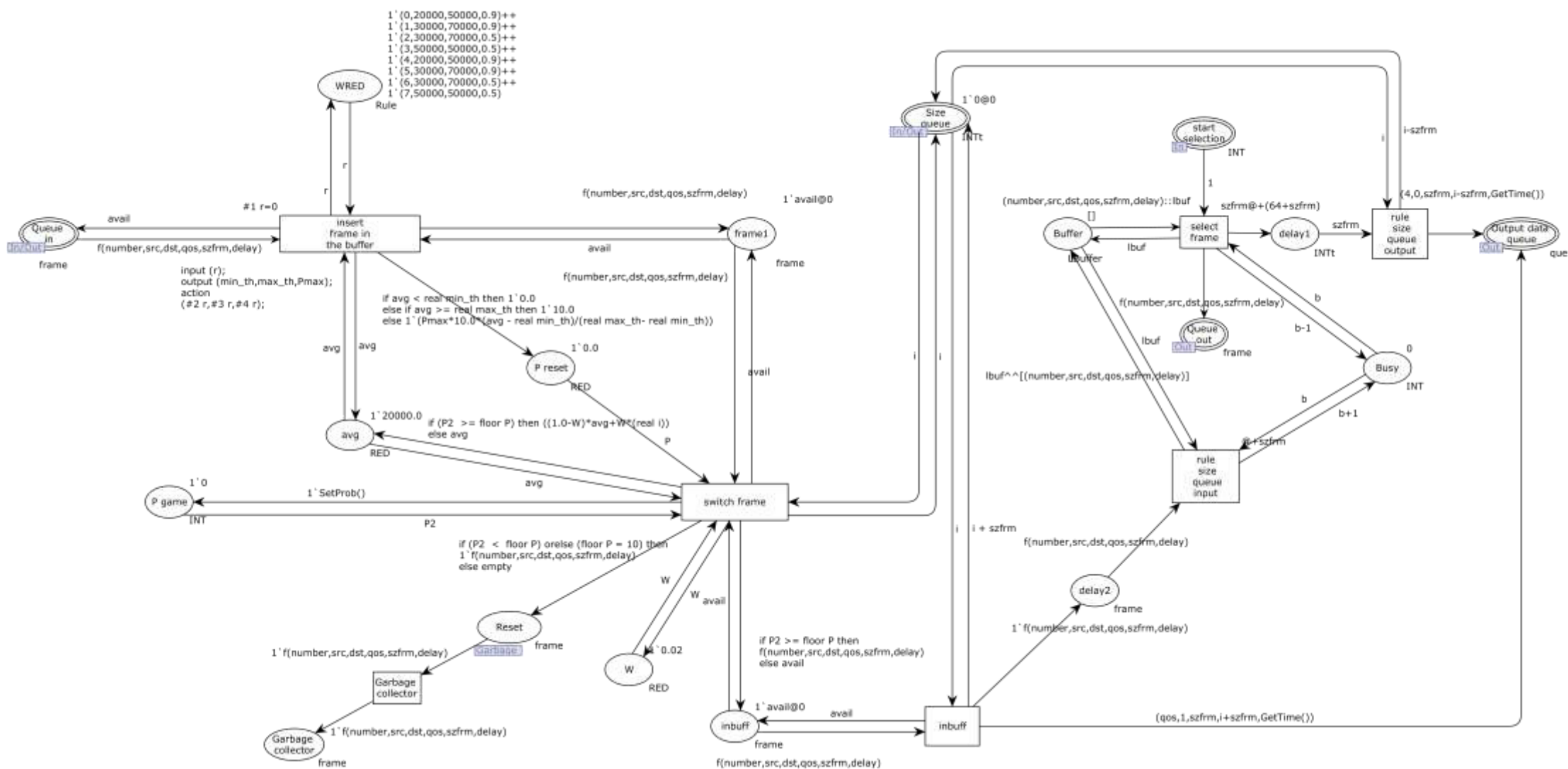


Рисунок 12 – Подсеть с полной буферизацией

Далее представлено описание всех позиций подсети *Queue*:

- *"Size queue"* – хранится значение размера очереди буфера коммутатора;
- *"Bufer In"* и *"Buffer Out"* – являются соответственно входом и выходом буфера выходного порта;
- *"Queue in"* и *"Queue out"* – являются соответственно входом и выходом очереди буфера выходного порта;
- *"Start selection"* – является сигналом начала выборки кадров из очереди буфера выходного порта.
- *Queue 4 in* – входная позиция подсети цвета frame, вход очереди буфера выходного порта;
- *Queue out* - выходная позиция подсети цвета frame, выход очереди буфера выходного порта;
- *Bufer* - очередь буфера;
- *Garbage Collector* - сборщик мусора, хранит маркеры, отброшенные соответствующим кадром;
- *Reset* – позиция, которая нужна для хранения маркера кадра, отбрасываемого в корзину;
- *Size queue* – маркер, отражающий текущий размер очереди буфера;
- *Start selection* - сигнал выборки кадра из очереди;
- *Output data queue* – маркер, отмечающий значение и время изменения очереди буфера;
- *W* - весовой коэффициент очереди;
- *avg* - среднее значение очереди буфера;
- *P reset* - вероятность отбрасывания пакетов, устанавливая функции сброса кадров при текущем значении средней очереди;

- *P game* – вероятность сброса пакета;
- *WRED* – содержит маркеры, значение цветов которых описание до 8 функций сброса кадров в алгоритме *WRED*;
- *frame1* – позиция для хранения маркера кадра перед принятием решения о его сбросе-продвижении.
- *inbuff* – позиция для хранения маркера кадра, направляемого в буфер;
- *delay2* – позиция, в которой находятся маркеры кадров, записываемые в очередь;
- *delay1* – позиция, в которой находятся маркеры во время передачи в выходной порт;
- *Busy* – маркер, отмечающий количество кадров в очереди готовых к чтению.

Далее представлено описание всех переходов подсети *Queue*:

- *insert frame in the buffer select frame* – вычисляет текущее значение средней скорости *avg* и вероятность сброса кадра при заданной функции сброса в соответствии выбранной ф-ей сброса кадра алгоритма *WRED*;
- *switch frame* – принимает значение о сбросе текущего кадра или направления его в буфер;
- *inbuff* – управляет подсчётами размеров очередей в битах;
- *rule size queue input* – записывает кадр в буфер FIFO в позиции буфер и управляющей подсчётом количества кадров в буфере в позиции *Busy*;
- *select frame* – считывает маркер головного кадра очереди FIFO и направляет его в выходной порт через позицию *Queue out*, кроме того управляет уменьшением размеров очередей, выраженных в кадрах;

– *rule size queue input* – управляет уменьшением размеров очереди в позиции *size queue* после окончания чтения очередного кадра из очереди, а также корректирует время и размер текущей очереди, записываемой в *Output data queue*.

Динамика работы подсети очереди.

Маркер, отмечающий очередной кадр поступает в позицию *Queue in* и вызывает срабатывание перехода *insert frame in the buffer*, который:

1. Вычисляет вероятность сброса кадра при текущем значении средней очереди из позиции *avg* по правилу:

```
if avg < real min_th then 1`0.0
else if avg >= real max_th then 1`10.0
else 1`(Pmax*10.0*(avg - real min_th)/(real max_th- real
min_th)) и записывает её в позицию P reset. Здесь значение min_th, max_th и Pmax вычисляются с помощью функции input:
```

```
input (r);
output (min_th,max_th,Pmax);
action
(#2 r,#3 r,#4 r);
```

В данной функции переменная *r* определяет тип функций сброса, описанных маркерами в позиции *WRED*. Для примера на рисунке 9 показаны следующие описания:

```
1`(0,20000,50000,0.9)++
1`(1,30000,70000,0.9)++
1`(2,30000,70000,0.5)++
1`(3,50000,50000,0.5)++
1`(4,20000,50000,0.9)++
1`(5,30000,70000,0.9)++
1`(6,30000,70000,0.5)++
1`(7,50000,50000,0.5)
```

Конкретное значение переменной *r* определяется предикатом срабатывания перехода *insert frame in the buffer*. На рисунке 9 указано

значение перехода #1 $r=0$, т.е. выбирается описание функции сброса, в первой позиции которого написано 0, т.е. модель использует только одну функцию сброса, что характерно для алгоритма RED. Алгоритм WRED будет реализовываться при предикате #1 $r=qos$, т.е. модель будет использовать каждый раз функцию сброса, соответствующую параметру приоритета кадра.

2. Передаёт кадр для дальнейшей обработки в позицию *frame1*.

Наличие маркеров в позициях *P reset* и *frame1* вызывает срабатывание перехода *switch frame*, который выполняет функции либо сброса кадра, находящегося в позиции *frame1* с заданной вероятностью *P*, либо в позицию *Reset* для последующего направления в корзину *Garbage collector*, накапливающую сброшенные кадры, либо передачи в позицию *inbuff* в противном случае. Условия выбора направления и передачи задаются соответствующими формулами:

– для сброса

```
if (P2 < floor P) or else (floor P = 10) then
  l`f(number,src,dst,qos,szfrm,delay)
else empty ;
```

– для записи в буфер

```
if P2 >= floor P then
  f(number,src,dst,qos,szfrm,delay)
else avail;
```

При этом переменная *P2* задаётся цветом маркера позиции и изменяется равновероятно случайным образом от 0 до 9 с помощью функции *I`SetProb()*.

Одновременно вычисляется новое значение средней скорости, изменяя цвет маркера в позиции *avg* по формуле:

```
if (P2 >= floor P) then ((1.0-W)*avg+W*(real i))
else avg
```

Маркер кадра, записываемого в буфер, в позиции *inbuff* вызывает срабатывание перехода *inbuff*, который увеличивает значение цвета размера буфера на длину кадра *szfrm*, что соответствует резервированию требуемой области памяти буфера. Но пока кадр не запишется в буфер, его нельзя использовать, поэтому маркер передаётся во временную позицию *delay2*, из которой он будет поставлен в очередь FIFO в позиции *Buffer* при срабатывании перехода *rule size queue input*, происходящей с задержкой равной $@+szfrm$ по формуле:

$$lbuf^{^}[(number,src,dst,qos,szfrm,delay)]$$

Момент записи маркера в позицию *Buffer* означает готовность его к передаче в выходной порт, что отмечается увеличением значения цвета маркера в позиции *Busy* (подсчёт количества готовых к обработке кадров в буфере).

Чтение информации из буфера осуществляется поступлением маркера в позицию *start selection*, по которому срабатывает переход *select frame*, читающий головной маркер из очереди FIFO по формуле:

$$(number,src,dst,qos,szfrm,delay)::lbuf$$

и передающей маркер в выходную позицию подсети *Queue out*. Одновременно с этим позиция *delay1* с задержкой, равной $szfrm@+(64+szfrm)$ передаётся длина кадра, которая переходом *rulesiz queue output*, который срабатывая:

а) изменяет, уменьшает размер очереди в позиции *Size queue*;

б) записывает в позицию *Output data queue* информацию о величине и моменте изменения очереди $(4,0,szfrm,i-szfrm,GetTime())$, которая обрабатывается монитором.

3.4 Описание универсальной модели алгоритмов управления очередью RED – WRED при буферизации «налету»

В разделе 3.3 рассмотрен алгоритм управления очередью RED – WRED для классического Ethernet коммутатора, выполняющего полную буферизацию передаваемого трафика. В целях повышения производительности коммутатора и сокращения задержек применяются коммутаторы, выполняющие коммутацию кадров «налету», в которых коммутатор считывает адрес назначения и продвигает кадр в выходной порт.

Следует отметить, что при занятости порта назначения в режиме «налету» может буферизироваться некоторый кадр, вплоть до его полного сохранения, поэтому режим можно рассматривать как частный случай первого режима (с полной буферизацией). Рисунок подсети *Queue* для коммутации «налету» представлен на рисунке 13.

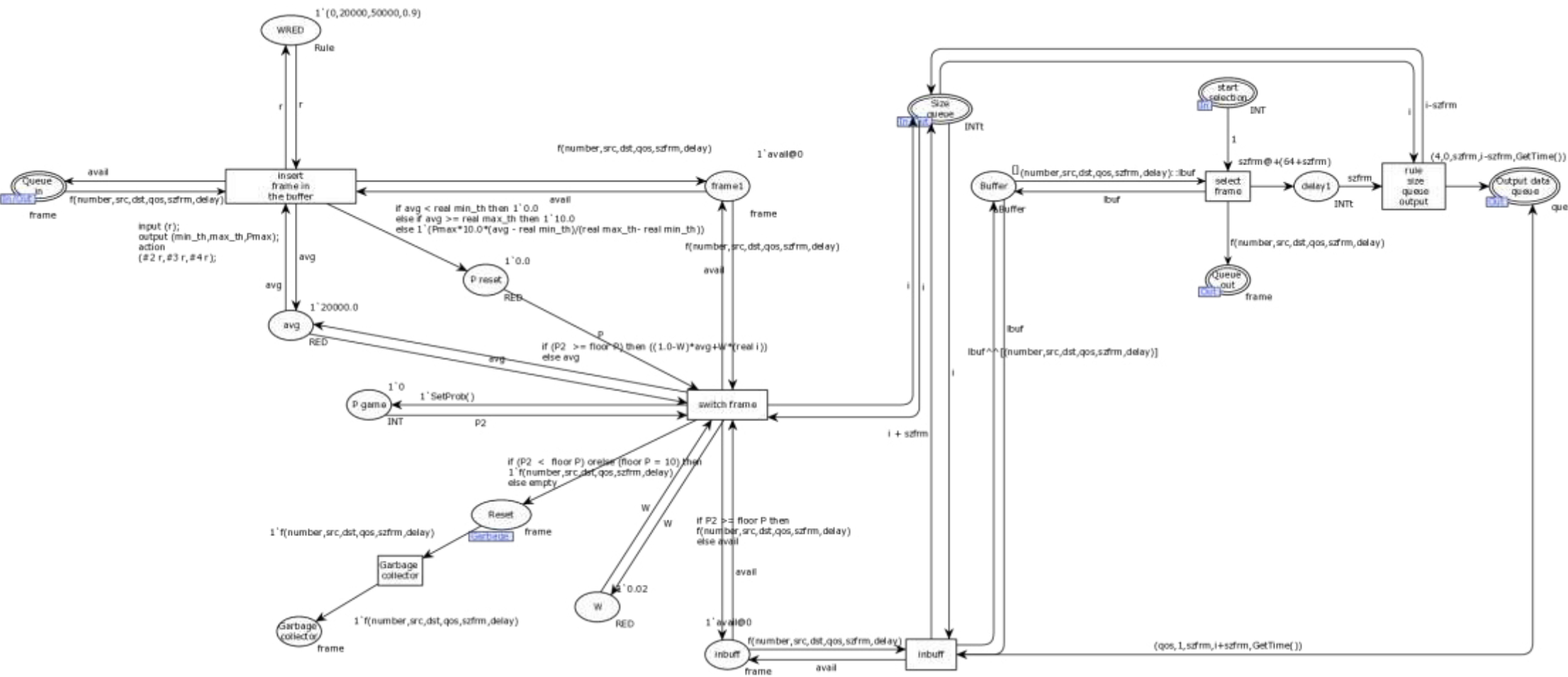


Рисунок 13 – RED – WRED с буферизацией «налету»

Из рисунка видно, что большая часть позиций и состояний элементов сети совпадают (см. раздел 3.3).

Исключены позиции *delay2* и *Busy*, а также переход *rule size queue input*, т.к. управление размером очереди в позиции *Size queue* выполняет непосредственно переход *input*, который выполняет также запись кадра, выбранного схемой управления, отобранного переходом *select frame* для записи непосредственно в очередь, формируемую в позиции *Buffer*.

Позиция *Size queue* используется также для указания готовности передачи помещённых в буфер кадров (рисунке 12 функция готовности формируется в позиции *rule size queue input*). Поэтому чтение из буфера становится возможным до окончания записи кадра в очередь.

4 ЭКСПЕРИМЕНТАЛЬНОЕ ИССЛЕДОВАНИЕ АЛГОРИТМОВ УПРАВЛЕНИЯ ОЧЕРЕДЯМИ RED – WRED

4.1 Исходные данные для моделирования.

Целью экспериментального исследования разработанной модели алгоритмов управления очередями RED – WRED является:

- Проверка правильности функционирования модели;
- Проверка функциональных возможности модели по оценки вероятностно-временных характеристик передачи трафика и очереди.

В качестве входного трафика использовался запись множества маркеров, моделирующих потока кадров Ethernet, записанная в файл с помощью специальной сети Петри. Этот поток представляет собой комбинацию потока кадров, и следующих через временные интервалы, распределенные по экспоненциальному закону со средним значением периода поступления, и квазирегулярный поток с редкими кадрами максимальной длительности.



Рисунок 14 – Гистограмма распределения интервалов между моментами поступления кадров

Коэффициент загрузки канала равен 0.8. Модельное время выражено в тактовых интервалах T_I , каждый из которых равен 100 нс для Ethernet или 10 нс для Fast Ethernet. Случайная длина кадров имеет следующее распределение, соответствующее реальным сетям: по 25 % от всех кадров, имеют минимальную и максимальную длину (соответственно $T_I \cdot 512$ и 12176), длина остальных равномерно распределена в оставшемся диапазоне (см. рисунок 15).

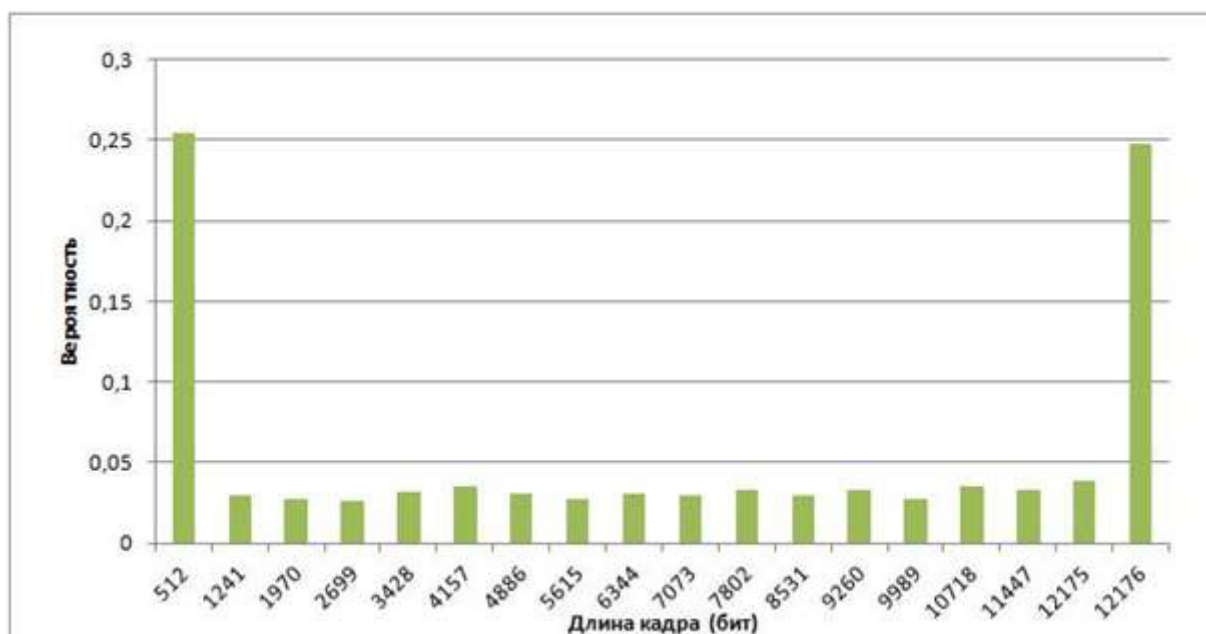


Рисунок 15 — Гистограмма распределения длин кадров сетевого трафика

Формируется 4 класса трафика, кадры имеют метки QoS от 1, 3, 5 и 7 с расчетной вероятностью поступления 0.4, 0.3, 0.2 и 0.1 соответственно (гистограмма распределения вероятности кадров различного класса приведена на рисунке 16).

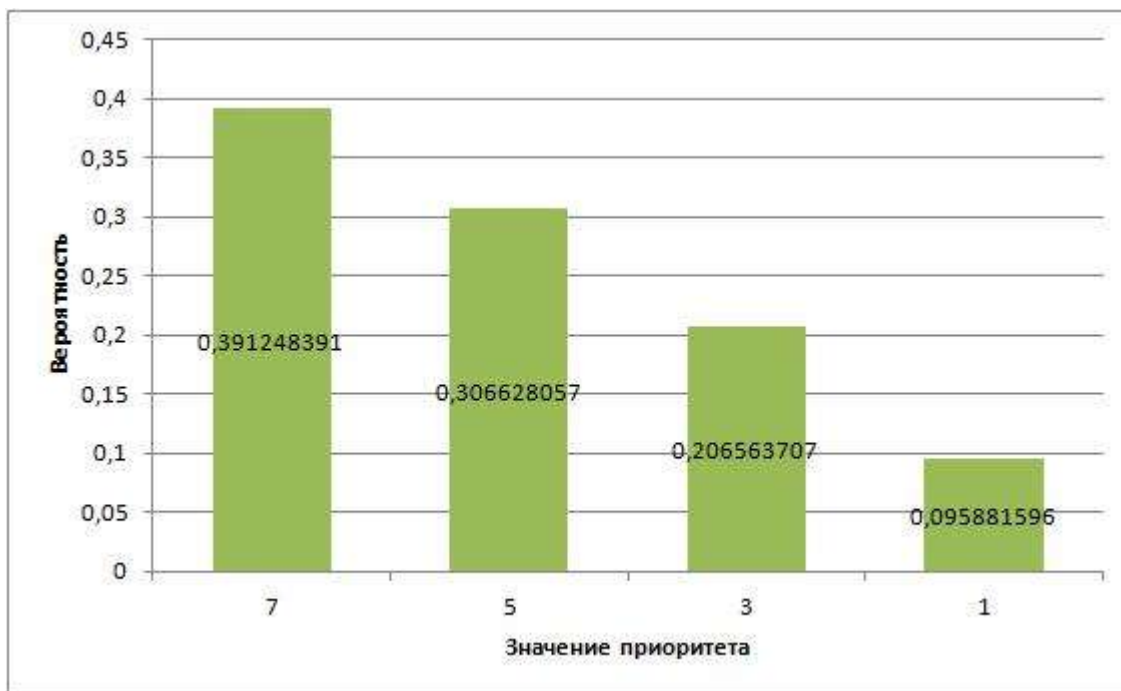


Рисунок 16 — Гистограмма распределения значений приоритетов кадров трафика

4.2 Обработка результатов экспериментов

В ходе проведения эксперимента создается файловая структура, пример которой представлен на рисунках 17-18.

Входные данные представляются файлом с именем traffic_3335.txt, который предварительно генерируется с помощью моделей, описанных в п.3.3-3.4.

WRED			
Имя	Дата изменения	Тип	Размер
output	04.06.2020 19:25	Папка с файлами	
traffic_3335.txt	07.06.2018 11:45	Текстовый докум...	140 КБ
WRED11.cpn	01.06.2020 13:56	Coloured Petri Net	296 КБ
WRED22.cpn	03.06.2020 13:54	Coloured Petri Net	297 КБ

Рисунок 17 – Пример файловой структуры, модель WRED1

WRED > output				Поис
Имя	Дата изменения	Тип	Размер	
New_Monitor.txt	04.06.2020 21:10	Текстовый докум...	0 КБ	
QueueWRR_band.txt	04.06.2020 21:10	Текстовый докум...	0 КБ	
WRR_band_2_3335.txt	04.06.2020 21:10	Текстовый докум...	0 КБ	

Рисунок 18 – Выходные файлы

Пример элементов этого файла, которые описывают каждый из подаваемых в модель маркеров (кадров), показан ниже:

```

1`(1,10,9,7,512, 296,0)++
1`(2,5,2,7,12096, 7389,0)++
1`(3,5,5,5,7952, 20178,0)++
1`(4,4,2,5,10848, 33939,0)++
1`(5,8,7,5,512, 34386,0)++
1`(6,8,9,5,6880, 71035,0)++
1`(7,1,4,3,512, 87014,0)++
1`(8,8,4,7,512, 92249,0)++
1`(9,9,1,1,512, 95918,0)++
1`(10,9,9,7,9544, 112645,0)++
1`(11,1,8,3,8840, 116330,0)++
1`(12,7,10,5,512, 139131,0)++
1`(13,9,4,7,512, 139717,0)++
1`(14,6,1,1,2640, 148585,0)++
1`(15,8,6,7,11160, 162034,0)++

```

В результате работы модели создается ряд выходных файлов, представленных на рисунке 18. Они размещены в папке *output*, а именно – *WRR_band_2_3335.txt*, *QueueWRR_band.txt*, *New_Monitor.txt*.

В файле *WRR_band_2_3335.txt* создается список параметров маркеров, покидающих модель, пример которой представлен ниже:

1,10,9,7,512, 296,0
 2,5,2,7,12096, 7389,0
 3,5,5,5,7952, 20178,0
 4,4,2,5,10848, 33939,0
 5,8,7,5,512, 34386,10561
 6,8,9,5,6880, 71035,0
 7,1,4,3,512, 87014,0

Значения элементов аналогичны описаниям входной с той разницей, что последним записана задержка в продвижении маркера через модель.

В файле QueueWRR_band.txt создается список элементов, описывающих изменение очередей:

7,1,512,20512,296
 4,0,512,20000,872
 7,1,12096,32096,7389
 4,0,12096,20000,19549
 5,1,7952,27952,20178
 4,0,7952,20000,28194
 5,1,10848,30848,33939
 5,1,512,31360,34386
 4,0,10848,20512,44851

Каждый элемент описывает:

- Номер очереди;
- Признак записи (1) или чтения (0) из очереди;
- Новое значение очереди;
- Время изменения очереди.

В файле New_Monitor.txt***, пример которой представлен ниже:

41,3,4,5,512, 397916,0
 47,6,2,5,10744, 428935,0

48,4,9,5,3576, 429169,0
 49,3,4,3,12176, 432507,0
 50,9,1,5,512, 436709,0
 51,5,4,5,11288, 444648,0
 53,1,1,1,12176, 465220,0
 56,8,8,7,12176, 489173,0
 57,1,10,3,512, 490573,0

Обработка выходных данных выполняется с помощью пакета MS Excel в следующей последовательности:

1. Загрузка файла в соответствующую книгу, с установками, показанными на рисунках 19-20.

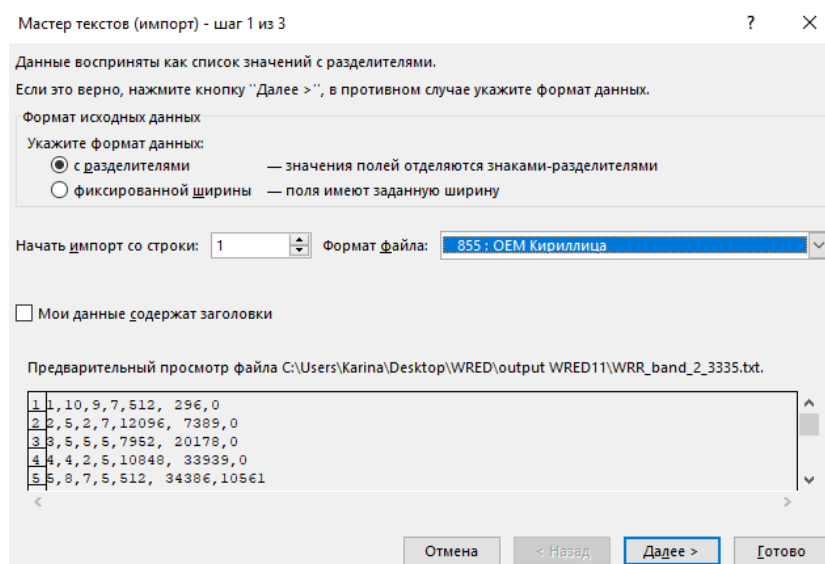


Рисунок 19 – Загрузка файла, шаг 1

Мастер текстов (импорт) - шаг 2 из 3

В этом диалоговом окне можно установить разделители для текстовых данных. Результат выводится в окне образца разбора.

Символом-разделителем является:

☒ знак табуляции

☐ точка с запятой

☒ запятая

☐ пробел

☐ другой:

☐ Считать последовательные разделители одним

Ограничитель строк:

Образец разбора данных

1	10	9	7	512	296	0
2	5	2	7	12096	7389	0
3	5	5	5	7952	20178	0
4	4	2	5	10848	33939	0
5	8	7	5	512	34386	10561

Отмена < Назад Далее > Готово

Рисунок 20 – Загрузка файла, шаг 2

В результате чего формируются стандартные листы, структура которых соответствует элементам приведенных выше файлов.

2. Селекция параметров маркеров, соответствующих QoS (1, 3, 5, 7) с помощью встроенных фильтров, как показано на рисунке 21, с последующим разделением по отдельным страницам.

	A	B	C	D	E	F	G	H	I
	1	10	9		512	296	0		
Сортировка по возрастанию					12096	7389	0		
Сортировка по убыванию					7952	20178	0		
Сортировка по цвету					10848	33939	0		
Удалить фильтр из столбца "7"					512	34386	10561		
Фильтр по цвету					6880	71035	0		
Числовые фильтры					512	87014	0		
					512	92249	0		
					512	95918	0		
Поиск					9544	112645	0		
(Выделить все)					8840	116330	6019		
☐ 1					512	139131	0		
☒ 3					512	139717	86		
☐ 5					2640	148585	0		
☐ 7					11160	162034	0		
					576	168914	4440		
					512	173769	321		
					9104	191798	0		
					512	198735	2327		
					960	210592	0		
					9896	233953	0		

OK Отмена

Рисунок 21 – Селекция параметров маркеров, соответствующих QoS «3»

3. Вычисление первичной статистики с использованием приложения «Анализ данных», пункт – «Описательная статистика». Пример результата показан на рисунке 22.

Описательная статистика	
Среднее	27423,66912
Стандартная ошибка	253,4967183
Медиана	22936
Мода	0
Стандартное отклонение	23084,90938
Дисперсия выборки	532913041,2
Экссесс	0,038714255
Асимметричность	0,786098763
Интервал	120416
Минимум	0
Максимум	120416
Сумма	227424488
Счет	8293

Рисунок 22 – Пример результатов вычисления первичной статистики

4. Построение гистограмм распределений кадров выполняется с использованием пакета «Анализ данных», пункт – «Гистограмма», как показано на рисунке 23.

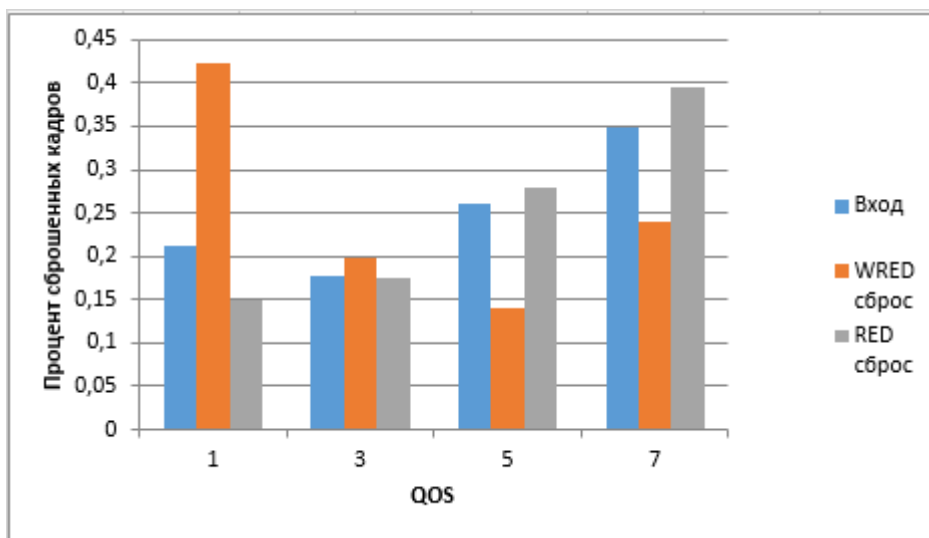


Рисунок 23 – Пример построения гистограмм

4.3 Анализ результатов экспериментов

Экспериментальные исследования разработанной модели проводились в следующем порядке:

1. Модель настраивалась на режим, исключающий сброс кадров (допущение бесконечной очереди) и определялись статистические характеристики очереди (см. таблицу 4.) Для настройки значения коэффициента W и начальной значения средней скорости avg устанавливались равным 0, а также заведомо большие значение $min_th = max_th = 200000$. Сброса кадров не наблюдается, статистические значения очереди представлены в таблице 4.

Полученные данные идентифицируем как исходные.

Таблица 4 - Статистические значения очереди

Столбец 1	Исход
Среднее	29497,43731
Стандартная ошибка	267,3710532
Медиана	24352
Мода	0
Стандартное отклонение	24560,32714
Дисперсия выборки	603209669,2
Эксцесс	0,104195335
Асимметричность	0,794467385
Интервал	125720
Минимум	0
Максимум	125720
Сумма	248899376
Счет	8438

Профиль очереди показан на рисунке 24.

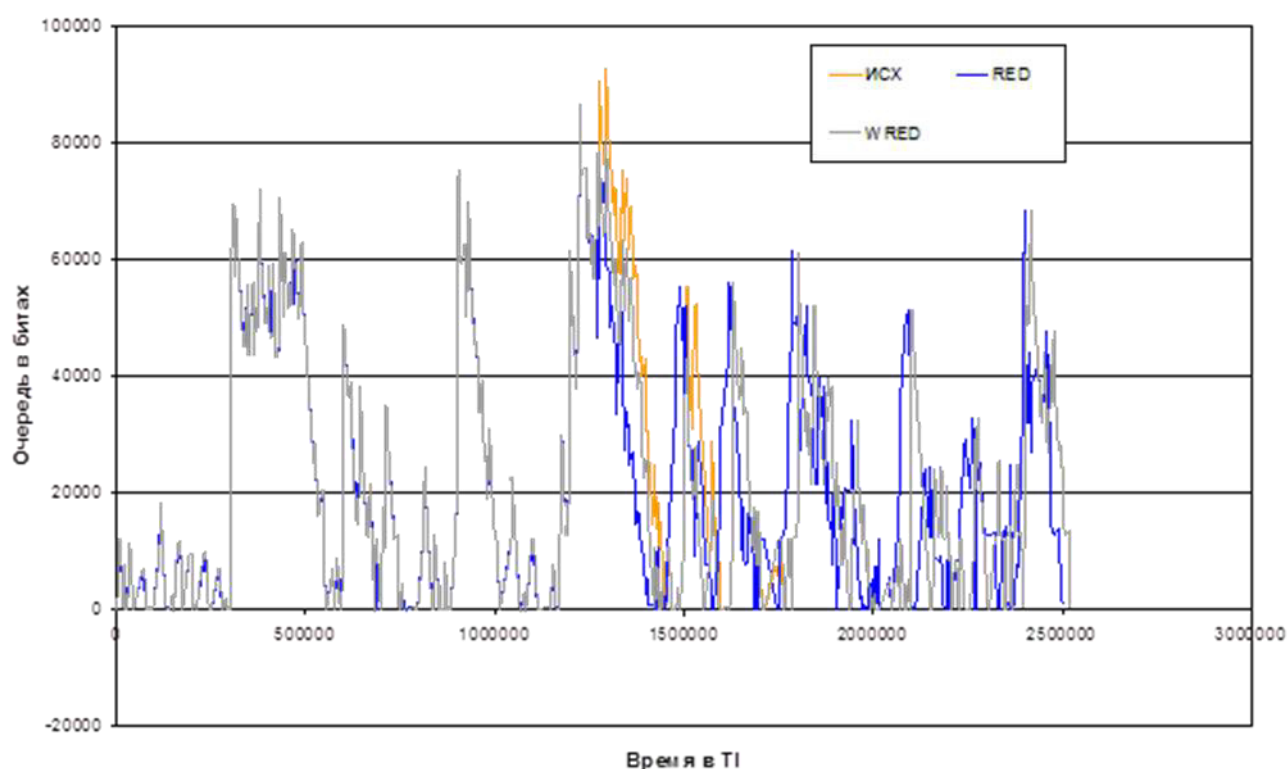


Рисунок 24 – Фрагмента профиля очереди для различных алгоритмов управления

2. На основании анализа статистических характеристик очереди, представленных в таблице 4, устанавливаем формы, которые представлены на рисунке 25.

Изм.	Лист	№ докум.	Подпись	Дата

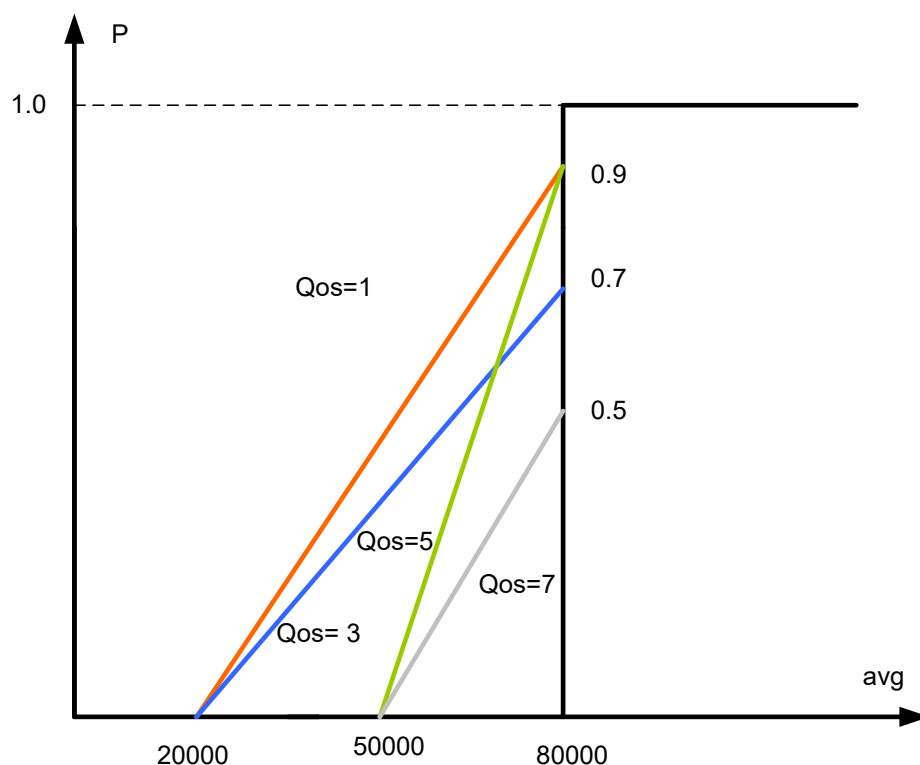


Рисунок 25 - Форма переходной характеристики принятия решений о сбросе пакета в алгоритмах RED и WRED

3. Настраиваем модель на реализацию алгоритма RED, выбрав переходную характеристику принятия решений о сбросе пакета, соответствующую $qos = 1$. Полученные статистические характеристики очереди приведены в таблице 5.

Таблица 5 - Статистические характеристики очереди

Столбец 1	RED
Среднее	25787,92814
Стандартная ошибка	245,6758837
Медиана	20968
Мода	0
Стандартное отклонение	22071,21293
Дисперсия выборки	487138440,2
Эксцесс	-0,143432146
Асимметричность	0,768062054
Интервал	113848
Минимум	0
Максимум	113848

Продолжение таблицы 5

Сумма	208134368
Счет	8071

Профиль очереди RED приведен в рисунке 26, значения по оси «У» соответствуют меткам qos сброшенных кадров.

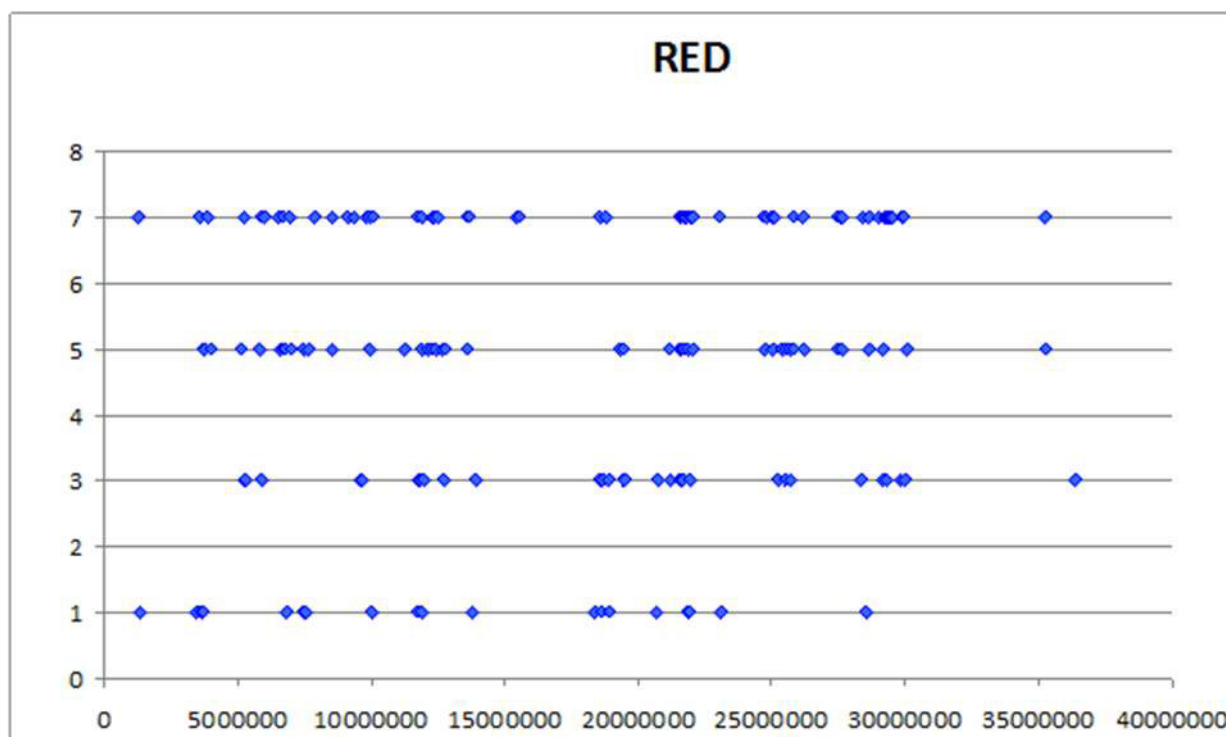


Рисунок 26 - Профиль потока сбрасываемых кадров в алгоритме RED

4. Настраиваем модель на реализацию алгоритма WRED, выбрав переходные характеристики принятия решений о сбросе пакета, соответствующие $qos = 1, 3, 5$ и 7 . Полученные статистические характеристики очереди приведены в таблице 6, профиль очереди WRED приведен в рисунке 27.

Таблица 6 - Полученные статистические характеристики очереди

Столбец1	WRED
Среднее	27423,66912
Стандартная ошибка	253,4967183
Медиана	22936
Мода	0
Стандартное отклонение	23084,90938
Дисперсия выборки	532913041,2
Эксцесс	0,038714255
Асимметричность	0,786098763
Интервал	120416
Минимум	0
Максимум	120416
Сумма	227424488
Счет	8293

Профиль очереди WRED приведен в рисунке 27, значения по оси «У» соответствуют меткам qos.

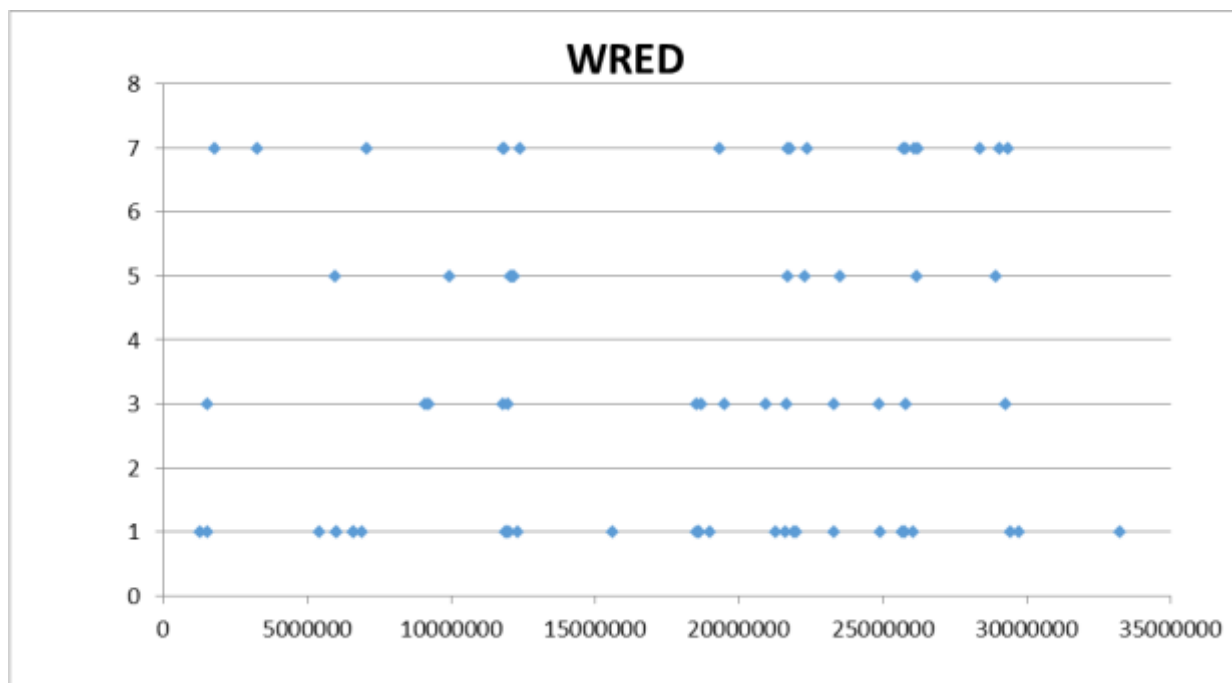


Рисунок 27 - Профиль потока сбрасываемых кадров в алгоритме WRED

5. Проведем сравнительный анализ доли кадров различных уровней Qos, попавших в очередь и сброшенных для различных алгоритмов управления. Статистика приведена в таблицах 7-8 и рисунках 28-29.

Таблица 7 - Статистика

QOS	Исход		RED очередь		WRED очередь	
	Кол.	Доля	Кол.	Доля	Кол.	Доля
1	895	0,212186	869	0,215206	864	0,208394
3	747	0,177098	716	0,177316	733	0,176797
5	1103	0,261498	1052	0,260525	1093	0,263628
7	1473	0,349218	1401	0,346954	1456	0,351182
	4218	1	4038	1	4146	1

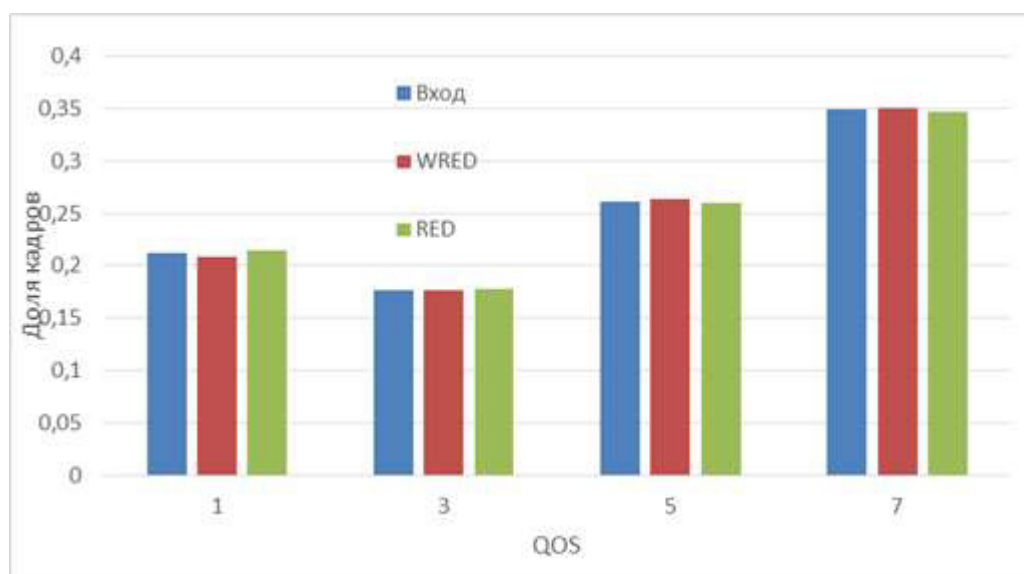


Рисунок 28 - Доля кадров различных уровней Qos, попавших в очередь.

Таблица 8 - Статистика

QOS	Исход		RED сброс		WRED сброс	
	Кол.	Доля	Кол.	Доля	Кол.	Доля
1	895	0,212186	27	0,148352	30	0,422535
3	747	0,177098	32	0,175824	14	0,197183
5	1103	0,261498	51	0,28022	10	0,140845
7	1473	0,349218	72	0,395604	17	0,239437
Всего	4218	1	182	1	71	1

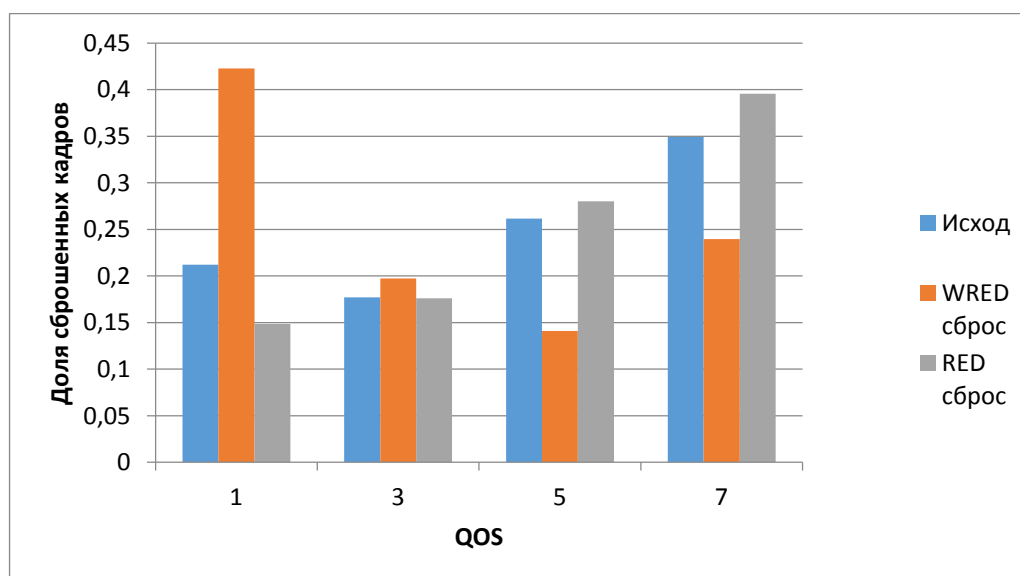


Рисунок 29 - Доля кадров различных уровней Qos, попавших в очередь.

Резкое отличие доли сброшенных кадров различных уровней Qos в алгоритме WRED от доли в соответствующих кадров в исходном потоке подтверждает правильность работы модели.

5 БЕЗОПАСНОСТЬ ЖИЗНЕДЕЯТЕЛЬНОСТИ

5.1 Введение

За последние 15 лет компьютерная индустрия развилась очень сильно, широко шагнув в практически все сферы промышленности и быта. В настоящее время трудно представить себе жизнь без компьютера. Но вместе с тем всё острее встаёт вопрос о безопасности пользователя при работе за компьютером, ведь при длительном использовании — компьютер становится источником вредного воздействия на организм человека и, соответственно, источником профессиональных заболеваний.

Каждый пользователь должен знать о вредном воздействии машин на организм и необходимых мерах защиты, чтобы сохранить свое здоровье и успешно работать за компьютером. Конечно существует ряд способов, которые если не полностью избавляют от вредных факторов, то существенно уменьшают их пагубное воздействие на организм.

5.2 Требования к уровню шума и вибрации

Уровни шума на рабочих местах пользователей персональных компьютеров не должны превышать значений, установленных СанПиН 2.2.4/2.1.8.562-96 и составляют не более 50 дБА. На рабочих местах в помещениях для размещения шумных агрегатов уровень шума не должен превышать 75 дБА, а уровень вибрации в помещениях допустимых значений по СН 2.2.4/2.1.8.566-96 категория 3, тип «в».

Снизить уровень шума в помещениях можно использованием звукопоглощающих материалов с максимальными коэффициентами звукопоглощения в области частот 63-8000 Гц для отделки стен и потолка помещений. Дополнительный звукопоглощающий эффект создают однотонные занавески из плотной ткани, повешенные в складку на

расстоянии 15-20 см от ограждения. Ширина занавески должна быть в 2 раза больше ширины окна.

5.3 Требования к освещению помещений и рабочих мест

Важное место в комплексе мероприятий по созданию условий труда, работающих с ПЭВМ, занимает создание оптимальной световой среды, т.е. рациональная организация естественного и искусственного освещения помещения и рабочих мест:

- помещения, предназначенные для размещения рабочих мест пользователей персональных компьютеров, должны иметь естественное и искусственное освещение;
- естественное освещение помещений, предназначенных для размещения рабочих мест, должно осуществляться через световые проемы, ориентированные преимущественно на север и северо-восток;
- в случае иной ориентации световых проемов, необходимо предусматривать эффективные средства регулирования интенсивности естественного освещения;
- оконные проемы в помещениях, предназначенных для использования компьютеров, необходимо оборудовать устройствами регулирования интенсивности естественного освещения типа жалюзи, занавесей, внешних козырьков и др.

5.4 Требования к организации и оборудованию рабочих мест

Рабочие места с персональными компьютерами по отношению к световым проемам должны располагаться так, чтобы естественный свет падал сбоку, желательно слева.

Схемы размещения рабочих мест с персональными компьютерами должны учитывать расстояния между рабочими столами с мониторами: расстояние между боковыми поверхностями мониторов не менее 1,2 м, а расстояние между экраном монитора и тыльной частью другого монитора не менее 2,0 м.

Целесообразно применение столов, имеющих отдельную от основной столешницы специальную рабочую поверхность для размещения клавиатуры. Используются рабочие столы с регулируемой и нерегулируемой высотой рабочей поверхности. При отсутствии регулировки высота стола должна быть в пределах от 680 до 800 мм.

Глубина рабочей поверхности стола должна составлять 800 мм (допускаемая не менее 600 мм), ширина — соответственно 1 600 мм и 1 200 мм. Рабочая поверхность стола не должна иметь острых углов и краев, иметь матовую или полуматовую фактуру.

Рабочий стол должен иметь пространство для ног высотой не менее 600 мм, шириной — не менее 500 мм, глубиной на уровне колен — не менее 450 мм и на уровне вытянутых ног — не менее 650 мм.

Клавиатура должна располагаться на поверхности стола на расстоянии 100-300 мм от края, обращенного к пользователю.

Конструкция стула должна обеспечивать:

- ширину и глубину поверхности сиденья не менее 400 мм;
- поверхность сиденья с закругленным передним краем;
- регулировку высоты поверхности сиденья в пределах 400-550 мм и углом наклона вперед до 15 градусов и назад до 5 градусов.;
- высоту опорной поверхности спинки 300 ± 20 мм, ширину — не менее 380 мм и радиус кривизны горизонтальной плоскости 400 мм;
- угол наклона спинки в вертикальной плоскости в пределах 0 ± 30 градусов;

- регулировку расстояния спинки от переднего края сидения в пределах 260-400 мм;
- стационарные или съемные подлокотники длиной не менее 250 мм и шириной 50-70 мм;
- регулировку подлокотников по высоте над сиденьем в пределах 230 ± 30 мм и внутреннего расстояния между подлокотниками в пределах 350-500 мм.;
- поверхность сиденья, спинки и подлокотников должна быть полумягкой, с нескользящим неэлектризующимся, воздухопроницаемым покрытием, легко очищаемым от загрязнения.

Рабочее место должно быть оборудовано подставкой для ног, имеющей ширину не менее 300 мм, глубину не менее 400 мм, регулировку по высоте в пределах до 150 мм и по углу наклона опорной поверхности подставки до 20 град. Поверхность подставки должна быть рифленой и иметь по переднему краю бортик высотой 10 мм.

ЗАКЛЮЧЕНИЕ

В ходе выполнения данной выпускной квалификационной работы была разработана модель взвешенного алгоритма произвольного раннего обслуживания очередями (WRED) в коммутаторе Ethernet, обеспечивающая возможность исследования вероятностных характеристик продвигаемого трафика в зависимости от интенсивности нагрузки и структуры поступающего трафика, а также была обеспечена возможность моделирования алгоритмов вероятностного заблаговременного определения перезагрузки (RED) и простого пассивного алгоритма управления очередями коммутаторов (Tail Drop).

					ПГУ 2.090301.14.001 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		68

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Олифер В.Г., Олифер Н.А. Компьютерные сети. Принципы, технологии, протоколы: Учебник для вузов. 3-е изд. – СПб - Питер, 2008, // с. 958.
2. В.Г. Олифер, Н.А Олифер. Компьютерные сети. Принципы, технологии, протоколы. 4-е изд. - СПб: Изд-во "Питер", 2010, // С. 943.
3. Яновский, Г.Г. Качество обслуживания в сетях IP. / Г.Г. Яновский. // Вестник связи. №1, 2008. – С. 65-74.
4. Д.А. Зайцев, Т.Р. Шмелева «Моделирование телекоммуникационных систем в CPN Tools. Учебное пособие по курсу «Математическое моделирование информационных систем» для подготовки магистров в отрасли связи. // Одесса, 2008.
5. Д.А. Зайцев. Сети Петри и моделирование систем, Методические указания к практическим занятиям и лабораторным работам для подготовки магистров по направлению «Телекоммуникации». // Одесса, 2006.
6. D.A. Zaitsev. Switched LAN simulation by colored Petri nets // Mathematics and Computers in Simulation, vol. 65, // 3, 2004. с. 245-249.
7. Котов В.Е. Сети Петри. - М.: Наука. Главная редакция физико-математической литературы, 1984.
8. Коваленко Т.Н. Модель активного управления очередями в распределенных инфокоммуникационных системах, представленная сетью Петри [Электронный ресурс] / Т.Н. Коваленко // Проблемы телекоммуникаций. – 2012. – № 2 (7). – С. 58 – 67. – Режим доступа к журн.: http://pt.journal.kh.ua/2012/2/1/122_kovalenko_queue.pdf.
9. Е. А. Кучерявый Управление трафиком и качество обслуживания в сети Интернет. // СПб: Изд-во "Наука и Техника", 2004, // с. 336.
10. Механов В.Б. Применение сетей Петри для моделирования механизмов обеспечения QoS в компьютерных сетях // Материалы

международного симпозиума “Новые информационные технологии и менеджмент качества (NIT&MQ’2010)” ФГУ ГНИИ ИТТ “Информика”. М. ЭГРИ, 2010

11. Домнин А.Л., Кизилев Е.А., Пушкарев В.А. Моделирование интеллектуальных коммутаторов Ethernet сетями Петри // Микроэлектроника и информатика - 2010. – М. МИЭТ, 2010.

12. В.В. Петров. Структура телетрафика и алгоритм обеспечения качества обслуживания при влиянии эффекта самоподобия // автореферат диссертации – М., 2004.

13. Гугель, Ю. В. Характеристики параметров «Качество обслуживания» опорной инфраструктуры научно-образовательной сети // Информатизация образования и науки. – 2009. – Вып. 4

14. Ш. Вегешна. Качество обслуживания в сетях IP. // М.: Изд-во <Вильямс>, 2003. // С. 368.

15. В.А. Гулиус Влияние дисциплины обслуживания заявок на производительность коммутатора [Электронный ресурс] http://model.exponenta.ru/cl_gva_05.html

16. Описание пакета CPN Tools [Электронный ресурс] - <http://cpntools.org/>.

17. В.Б. Механов. Моделирование сетями Петри алгоритмов обслуживания очередей // Материалы международного симпозиума «Новые информационные технологии и менеджмент качества (NIT&MQ'2011)» ФГУ ГНИИ ИТТ «Информика». // М.: ООО «Арт-Флеш», 2011. -С. 8-11.

18. Модель управления очередями на маршрутизаторах Т. Р. Велиева, А. В. Королькова, Д. С. Кулябов, Б. А. Сантуш Кафедра систем телекоммуникаций Российский университет дружбы народов ул. Миклухо-Маклая, д.6, Москва, 117198, Россия

19. А. В. Королькова, Д. С. Кулябов, А. И. Черноиванов Кафедра
систем телекоммуникаций Российский университет дружбы народов ул.
Миклухо–Маклая, 6, Москва, Россия, 117198

					ПГУ 2.090301.14.001 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		71

Приложение А Презентация

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА БАКАЛАВРА

Моделирование алгоритма WRED для управления очередями в коммутаторах

Автор работы: студентка группы 16вв2 Жидкова К.О
Научный руководитель: к.т.н., профессор Коннов Н.Н.

Цель работы

Цель:

- Разработка модели взвешенного алгоритма произвольного раннего обслуживания очередями (WRED) в коммутаторе Ethernet, обеспечивающая возможность исследования вероятностных характеристик продвигаемого трафика в зависимости от интенсивности нагрузки и структуры поступающего трафика.
- Обеспечение возможности моделирования алгоритмов вероятностного заблаговременного определения перезагрузки (RED) и простого пассивного алгоритма управления очередями коммутаторов (Tail Drop).

Изм.	Лист	№ докум.	Подпись	Дата

ПГУ 2.090301.14.001 ПЗ

Лист

72

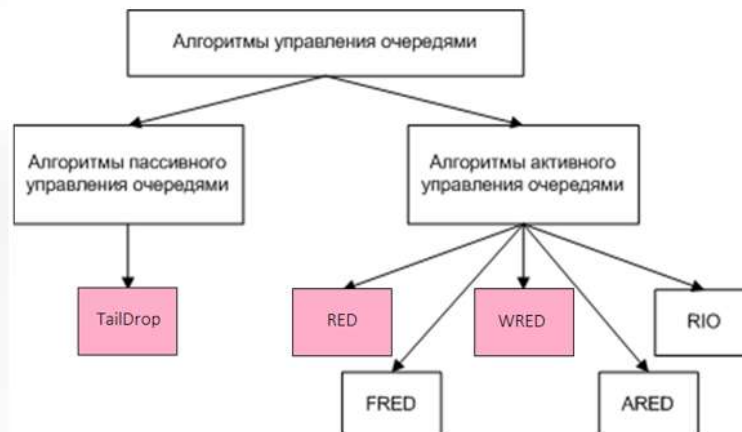
Что такое качество обслуживания QoS

Quality of Service (QoS) — технология предоставления различным классам трафика различных приоритетов в обслуживании — качества обслуживания.

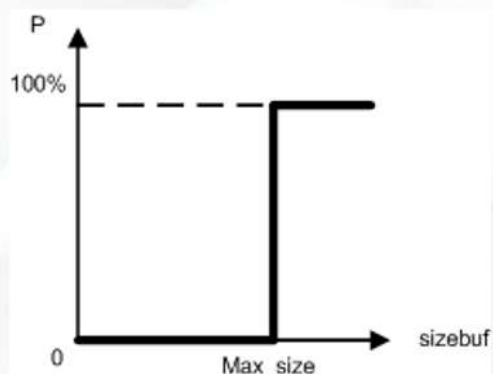
Механизмы реализации QoS:

- механизмы маркирования;
- механизмы управления очередями;
- механизмы обслуживания очередей;
- механизмы управления полосой пропускания.

Классификация алгоритмов управления очередями



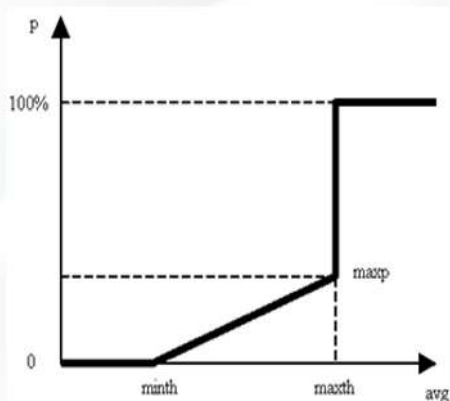
Алгоритм TailDrop



Зависимость вероятности сброса от
длины очереди

$$P_a = \begin{cases} 0, & \text{если } sizebuf < max_size \\ 1, & \text{если } sizebuf > max_size \end{cases}$$

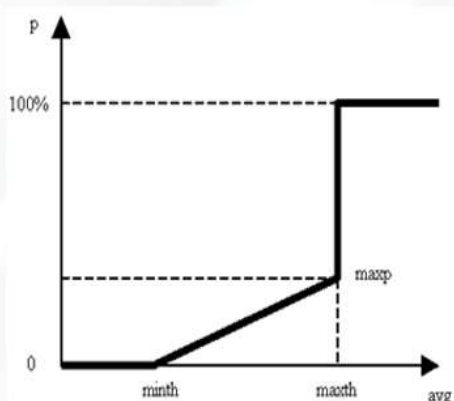
Алгоритм RED



Зависимость вероятности сброса от
усредненной длины очереди

$$P_a = \begin{cases} 0, & \text{если } avg < min_{th} \\ max_p * \frac{avg - min_{th}}{max_{th} - min_{th}}, & \text{если } min_{th} \leq avg < max_{th} \\ 1, & \text{если } avg \geq max_{th} \end{cases}$$

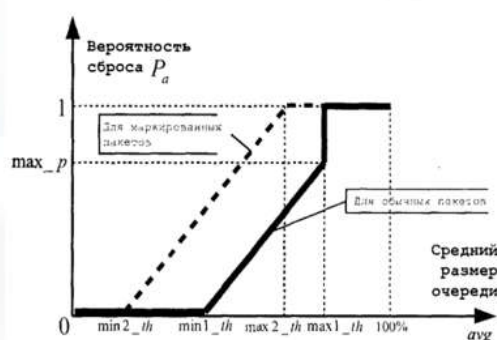
Алгоритм RED



Зависимость вероятности сброса от усредненной длины очереди

$$P_a = \begin{cases} 0, & \text{если } avg < min_{th} \\ max_p * \frac{avg - min_{th}}{max_{th} - min_{th}}, & \text{если } min_{th} \leq avg < max_{th} \\ 1, & \text{если } avg \geq max_{th} \end{cases}$$

Алгоритм WRED



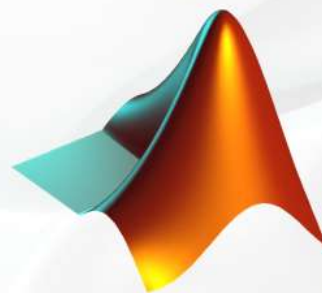
Зависимость вероятности отбрасывания пакета от значений средневзвешенной длины очереди

$$avg(k+1) = (1-w_{avg})avg(k) + w_{avg}avg(k)$$

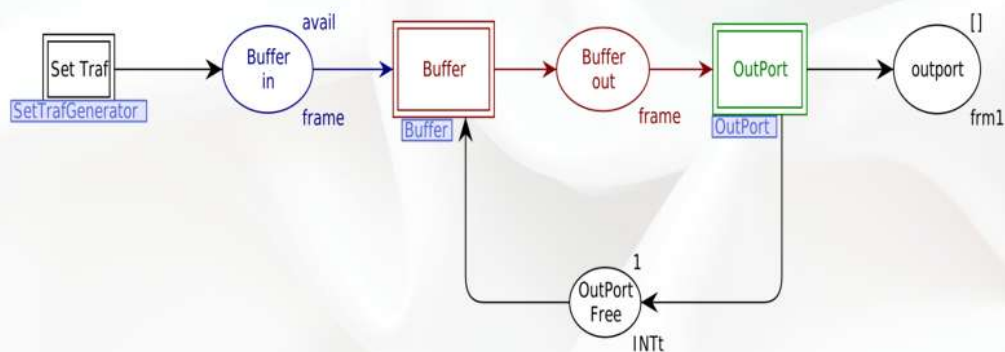
пока $avg < min_{th}$ - входящий пакет поступает в буфер
если $min_{th} < avg < max_{th}$ - вероятность отбрасывания растёт от 0 до pc
 pc – max значение вероятности отбрасывания пакетов в области между min_{th} и max_{th}
после достижения порога max_{th} все поступающие пакеты отбрасываются

Выбор аппарата для имитационного моделирования

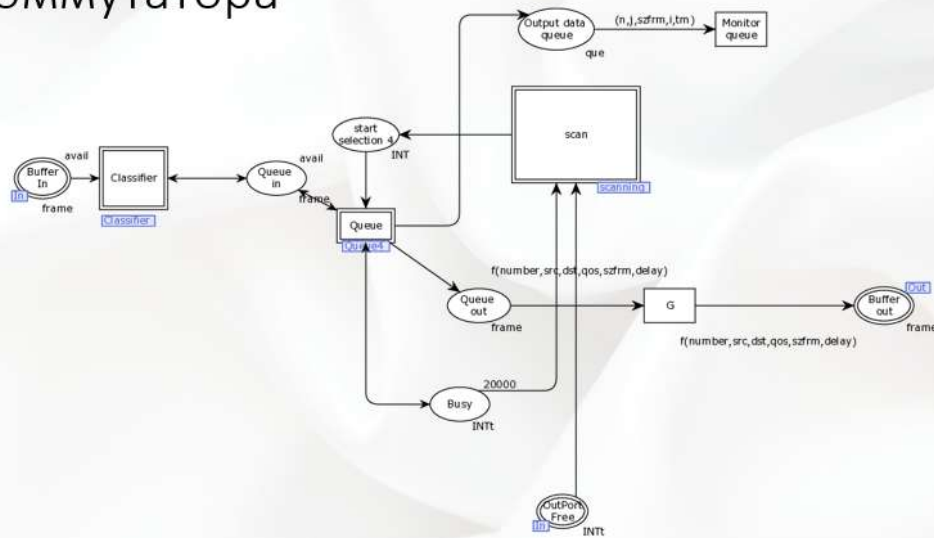
- MATLAB
- GPSS
- CPN Tools



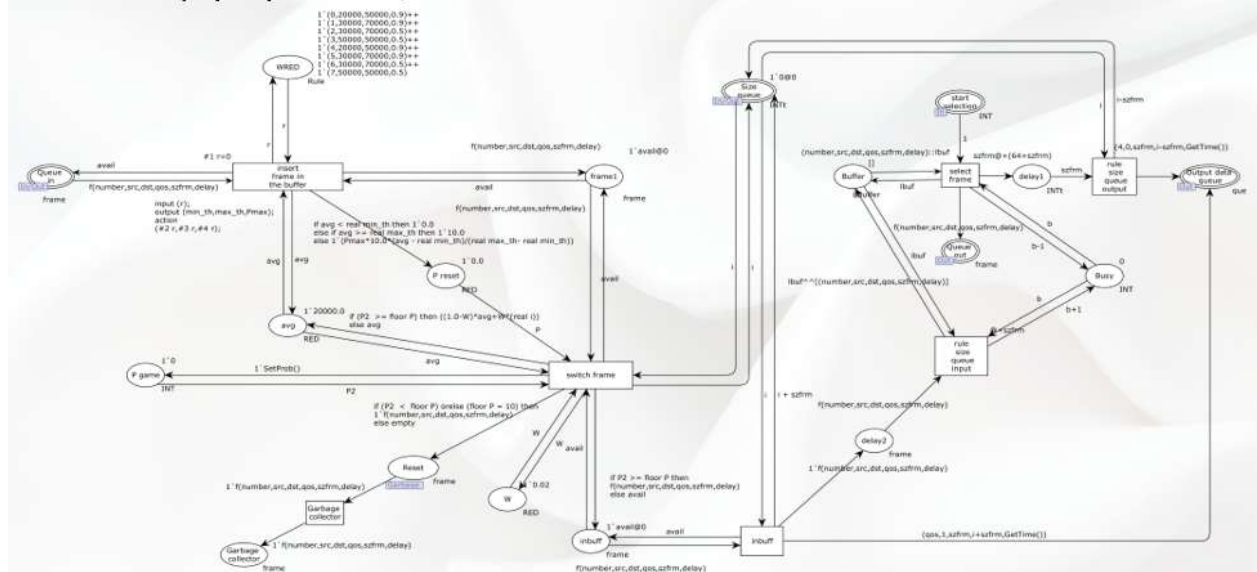
Обобщенная структура модели буфера коммутатора



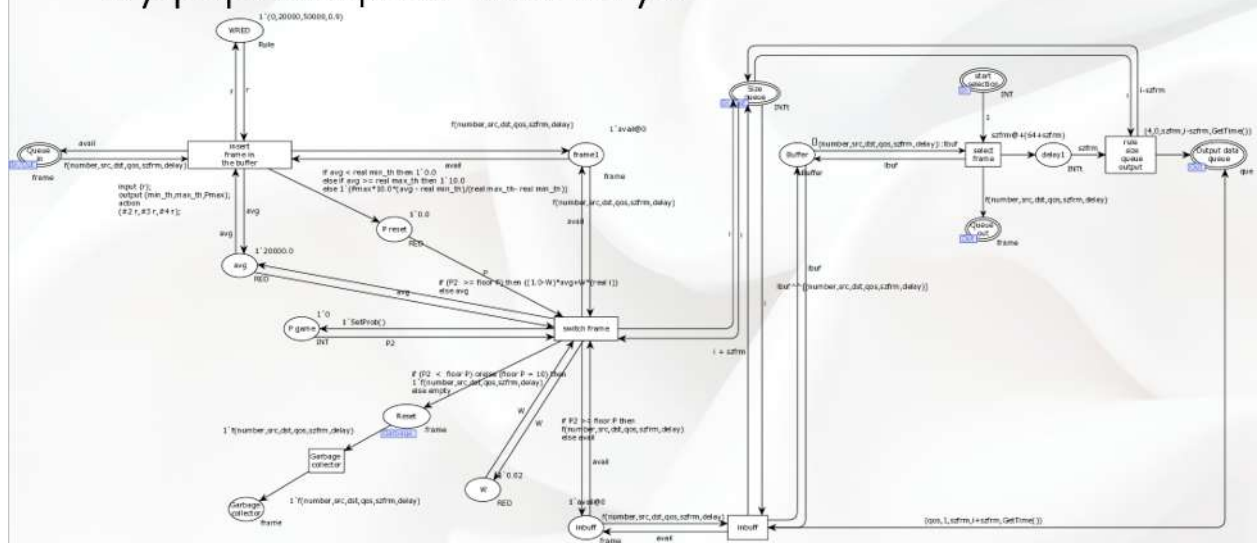
Модель буфера выходного порта коммутатора



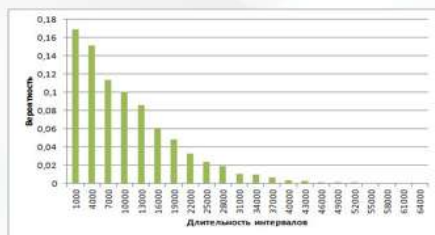
Модель алгоритма RED-WRED с полной буферизацией



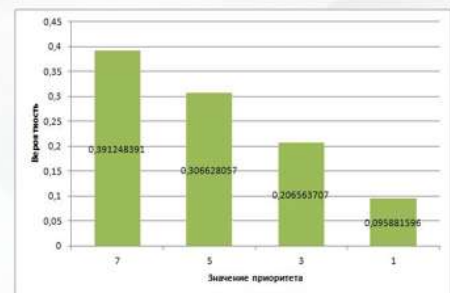
Модель алгоритма RED-WRED с буферизацией «налету»



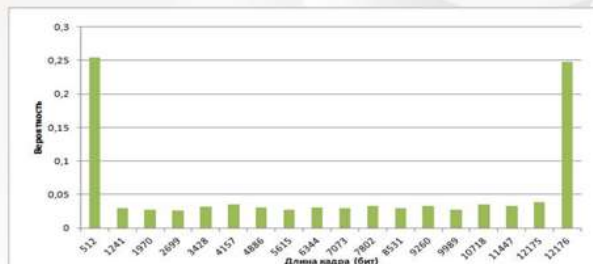
Экспериментальное исследование



Гистограмма распределения интервалов между моментами поступления кадров

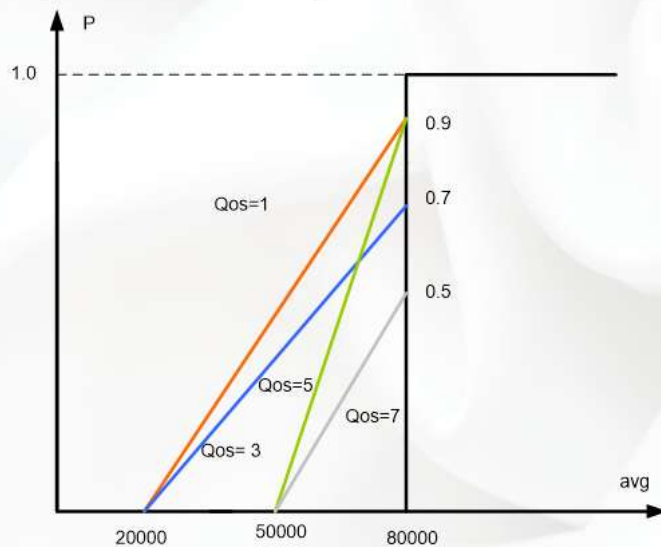


Гистограмма распределения значений приоритетов кадров трафика

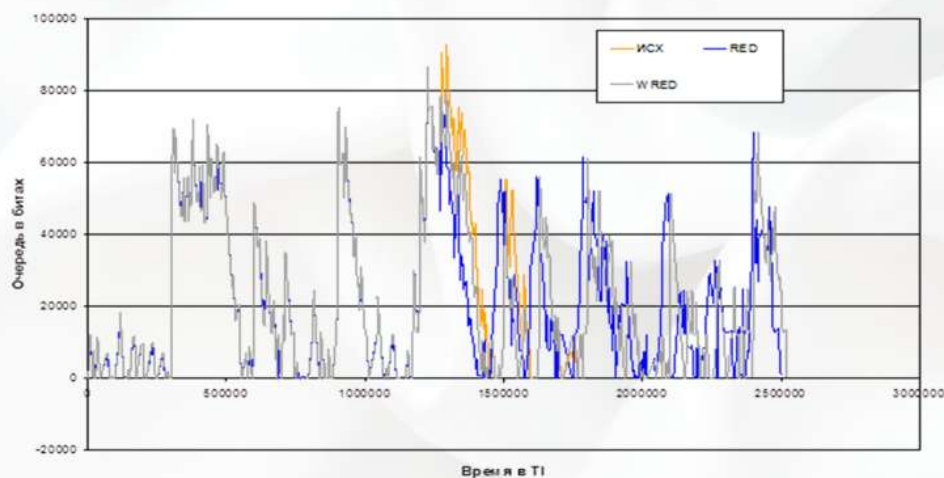


Гистограмма распределения длин кадров сетевого трафика

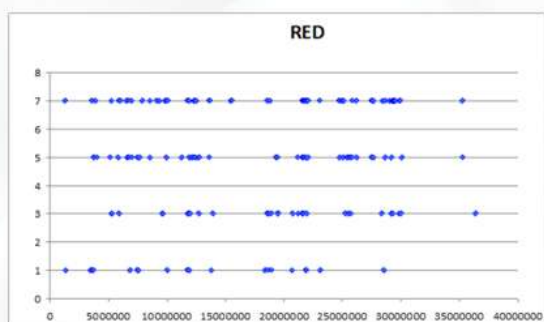
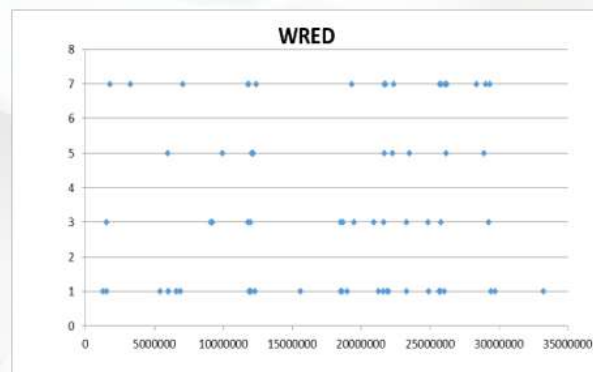
Форма переходной характеристики принятия решений о сбросе пакета в алгоритмах RED и WRED



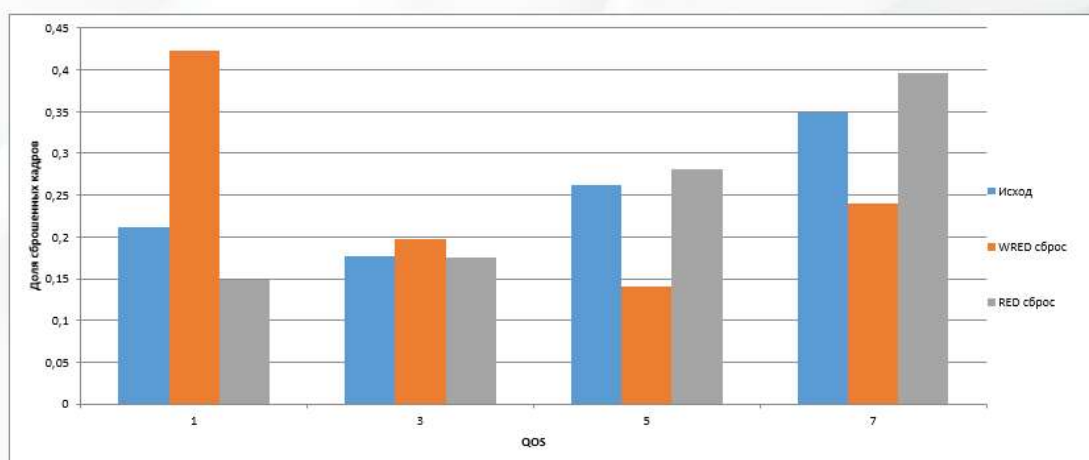
Фрагмент профиля очереди для различных алгоритмов управления

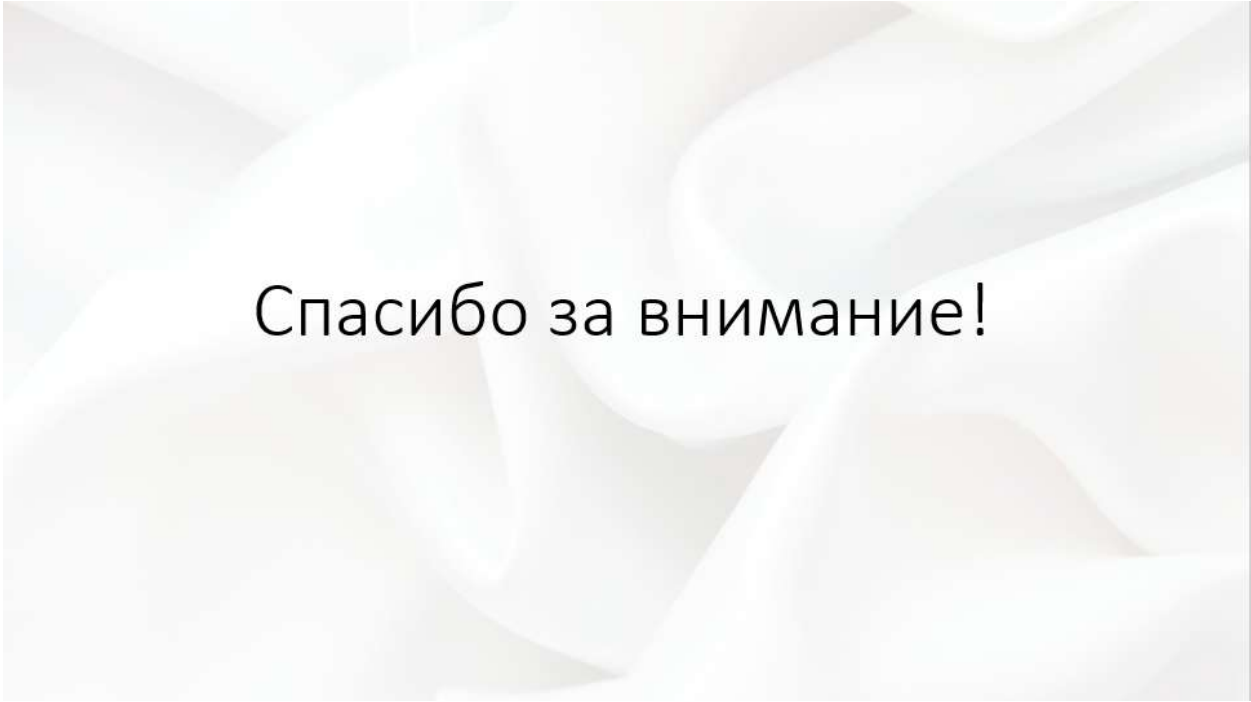


Профили потока сбрасываемых кадров в алгоритмах RED и WRED



Доля кадров различных уровней Qos, попавших в очередь





Спасибо за внимание!

Изм.	Лист	№ докум.	Подпись	Дата

ПГУ 2.090301.14.001 ПЗ

Лист

81